

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

**КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І  
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ**

«На правах рукопису»  
УДК УДК 004.05

«До захисту допущено»  
Завідувач кафедри СПСКС

\_\_\_\_\_  
(підпис) В.П.Тарасенко  
(ініціали, прізвище)  
“ ” \_\_\_\_\_ 2018р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

зі спеціальності 123 Комп'ютерна інженерія  
(Системне програмування)

на тему: Метод перетворення GL-моделей шляхом введення внутрішніх ребер з  
урахуванням попарних реберних циклів

Виконав (-ла): студент (-ка) II курсу, групи KB-62м

Примак Іван Костянтинович

\_\_\_\_\_  
(підпис)

Науковий керівник доцент, к.т.н. Романкевич В.О.

\_\_\_\_\_  
(підпис)

Рецензент \_\_\_\_\_

\_\_\_\_\_  
(підпис)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2018 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія

(Системне програмування)

ЗАТВЕРДЖУЮ

Завідувач кафедри СПСКС

\_\_\_\_\_ В.П.Тарасенко  
(підпис) (ініціали, прізвище)

«\_\_\_» \_\_\_\_\_ 2018р.

**ЗАВДАННЯ  
на магістерську дисертацію студенту  
Примаку Івану Костянтиновичу**

1. Тема дисертації: Метод перетворення GL-моделей шляхом введення внутрішніх ребер з урахуванням попарних реберних циклів, науковий керівник дисертації Романкевич Віталій Олексійович, доцент кафедри СПіСКС, к.т.н., доцент, затверджені наказом по університету від «22» березня 2018 р. №986-с
2. Термін подання студентом дисертації 11 травня 2018 р.
3. Об'єкт дослідження: моделі, що відображують поведінку відмовостійких багатопроцесорних систем, та методи їх створення та підтримки під час використання
4. Предмет дослідження: метод перетворення GL-моделей шляхом введення внутрішніх ребер з урахуванням попарних реберних циклів
5. Перелік завдань, які потрібно розробити
  - опис предметної області досліджень;
  - аналіз відомих методів створення GL-моделей;
  - метод перетворення GL-моделей шляхом введення внутрішніх ребер з урахуванням попарних реберних циклів;
  - аналіз критеріїв ефективності використання методу.

6. Перелік ілюстративного матеріалу

- Стабільні S-підмножини
- Побудова V-графу
- Побудова модифікованого графу
- V'-граф для моделі з попарними реберними циклами
- Конфігурації попарних реберних циклів
- Попарні реберні цикли в GL-моделі

7. Перелік публікацій

- “Method of modification of base graph-logical models“, 20-th International Conference SAIT 2018 (Київ, 21-24 травня 2018 р.);;
- “Метод перетворення GL-моделей шляхом введення внутрішніх ребер з урахуванням попарних реберних циклів”, X наукова конференція магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018 (Київ, 21-23 березня 2018 р.).

8. Дата видачі завдання 5 вересня 2016 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Вивчення літератури за тематикою проекту	20.10.2017	
2	Аналіз існуючих рішень	20.12.2017	
3	Підготовка матеріалів першого розділу магістерської дисертації	09.01.2018	
4	Підготовка матеріалів другого розділу магістерської дисертації	05.02.2018	
5	Підготовка матеріалів третього розділу магістерської дисертації	05.03.2018	
6	Підготовка графічної частини дипломного проекту	23.03.2018	
7	Оформлення документації дипломного проекту	20.04.2018	
8	Попередній розгляд магістерської дисертації на кафедрі	26.04.2018	

Студент

\_\_\_\_\_  
(підпис)

Науковий керівник дисертації

\_\_\_\_\_  
(підпис)

Примак І.К.

(ініціали, прізвище)

Романкевич В.О.

(ініціали, прізвище)

## РЕФЕРАТ

**Актуальність теми.** Використання моделей при проектуванні складних систем вже певний час є загальною практикою. Такі моделі використовуються для проведення експериментів, за результатами яких можна оцінити кількісні характеристики, такі як рівень надійності і ефективність цих систем ще на етапах проектування, що дуже важливо у зв'язку з високою вартістю побудови таких систем. Створення нових та вдосконалення вже існуючих методів побудови таких моделей з метою досягнення більшої швидкості визначення стану систем впродовж виконання статистичних тестів, що виконуються з метою визначення рівня надійності системи, спрощує та прискорює роботу спеціалістів, що використовують ці моделі у своїй праці.

**Об'єктом дослідження** є моделі, що відображують поведінку відмовостійких багатопроцесорних систем, та методи їх створення та підтримки під час використання.

**Предметом дослідження** є метод перетворення GL-моделей шляхом введення внутрішніх ребер з урахуванням попарних реберних циклів.

**Мета роботи:** аналіз існуючих методів перетворення GL-моделей, створення методу, що пропонує рішення для вибору додаткових ребер та реберних функцій враховуючи попарні реберні цикли та їх можливі комбінації.

**Наукова новизна** полягає в наступному:

1. Запропоновано метод перетворення GL-моделей шляхом введення внутрішніх ребер з урахуванням попарних реберних циклів.
2. Виконана оцінка мінімальної кількості додаткових внутрішніх ребер, необхідних для блокування втрати зв'язності графу GL-моделі з попарними реберними циклами.

**Практична цінність** отриманих в роботі результатів полягає в тому, що запропонований методи дають змогу створювати GL-моделі, які коректно відображують поведінку систем у разі утворення попарних реберних циклів, а також спрощує побудову таких моделей.

**Апробація роботи.** Основні положення і результати роботи були представлені та обговорювались на науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018 (Київ, 21-23 березня 2018 р.) та міжнародній конференції «Системний аналіз та інформаційні технології» SAIT 2018.

**Структура та обсяг роботи.** Магістерська дисертація складається з вступу, трьох розділів та висновків.

*У вступі* подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі досліджень, показано наукову новизну отриманих результатів і практичну цінність роботи, наведено відомості про апробацію результатів і їхнє впровадження.

*У першому розділі* розглянуто існуючі методи побудови моделей відмовостійких багатопроцесорних обчислювальних систем, а також проведений аналіз, який дає змогу визначити основні переваги та недоліки цих методів.

*У другому розділі* наведено результати дослідження методу перетворення GL-моделей шляхом введення внутрішніх ребер та явище попарних реберних циклів у наборі блокованих множин.

*У третьому розділі* формулюється метод перетворення GL-моделей шляхом введення внутрішніх ребер з урахуванням попарних реберних циклів та надається алгоритм на його основі.

*У висновках* представлені результати проведеної роботи.

У *додатках* наведено схеми перетворення базових GL-моделей у небазові шляхом введення внутрішніх ребер, схеми побудови V-графу моделі та типи попарних реберних циклів.

Робота виконана на 81 аркуші, містить 6 додатків та посилання на список використаних літературних джерел з 19 найменувань. У роботі наведено 50 рисунків та 9 таблиць.

**Ключові слова:** відмовостійкі багатопроцесорні системи, надійність систем, математичне моделювання, графо-логічні моделі.

## ABSTRACT

**Relevance of the topic.** The use of models in the design of complex systems is a common practice for quite a sometime. Such models are used to carry out experiments that can be used to evaluate quantitative characteristics such as the level of reliability and efficiency of these systems at the design stages, which is very important due to the high cost of constructing such systems. Creation and improvement of existing methods of constructing such models in order to achieve a higher speed of determining the state of the systems simplifies and accelerates the work of specialists who use these models in their work.

**The object of the study** are the models that reflect the behavior of fault-tolerant multiprocessor systems and the methods of their creation and support during use.

**The subject of the study** is the method of converting GL-models by introducing internal edges taking into account pairwise edge cycles.

**Purpose:** the analysis of existing methods for converting GL models, the creation of a method that offers solutions for the selection of additional edges and edge functions, taking into account pairwise edge cycles and their possible combinations.

**The scientific novelty** is as follows:

1. The method of transforming GL-models by introducing internal edges taking into account pairwise edge cycles is proposed.
2. The estimation of the minimum number of additional internal edges required to prevent the loss of connectivity of the graph of the GL model with the pairwise edge cycles is performed.

**The practical value** of the results obtained in the work is that the proposed methods allow the creation of GL models that correctly reflect the behavior of systems in the case of pairwise edge cycles, and simplifies the construction of such models.

**Test work.** The main provisions and results of work were presented and discussed at the scientific conference of masters and postgraduate students «Прикладна математика та комп'ютинг» ПМК-2018 (Kyiv, 21-23 of March 2018) and at international conference “System Analysis and Information Technology” SAIT 2018.

**Structure and scope of work.** The master's dissertation consists of an introduction, three sections and conclusions.

*The introduction* gives a general description of the work, assesses the current state of the problem, substantiates the relevance of the research direction, formulates the purpose and objectives of the research, shows the scientific novelty of the results obtained and the practical value of the work, provides information on the approbation of the results and their implementation.

*The first section* examines the existing methods for building fault-tolerant multiprocessor computer systems, as well as analyzes that can identify the main advantages and disadvantages of these methods.

*The second section* presents the results of the study of the method of converting GL-models by introducing internal edges and the phenomenon of pairwise finer cycles in a set of blocked sets.

*In the third section*, the method of converting GL-models by introducing internal ribs taking into account pairwise edge cycles is formulated and an algorithm based on it is provided.

*The conclusion* presents the results of the work.

*The appendices* show the schemes of transformation of basic GL-models in non-base by introducing internal edges, schemes of construction of the V-graph model and types of pairwise edge cycles

The thesis is presented in 81 pages, it contains 6 appendixes and 20 references to the used information sources. 50 figures and 9 tables are given in the thesis.

**Keywords:** fault-tolerant multiprocessor systems, fault tolerance, mathematic modelling, graph-logical models.



## РЕФЕРАТ

**Актуальность темы.** Использование моделей при проектировании сложных систем уже некоторое время является общей практикой. Такие модели используются для проведения экспериментов, по результатам которых можно оценить количественные характеристики, такие как уровень надежности и эффективности этих систем еще на этапах проектирования, что очень важно в связи с высокой стоимостью построения таких систем. Создание и совершенствование уже существующих методов построения таких моделей с целью достижения большей скорости определения состояния систем упрощает и ускоряет работу специалистов, использующих эти модели в своей работе.

**Объектом исследования** является метод преобразования GL-моделей путем введения внутренних ребер с учетом парных реберных циклов.

**Предметом исследования** являются алгоритмы и аппаратно-программные средства формирования GL-модели и ее поддержки во время использования.

**Цель работы:** анализ существующих методов преобразования GL-моделей, создания метода, предлагает решения для выбора дополнительных ребер и реберных функций учитывая попарные реберные циклы и их возможные комбинации.

**Научная новизна** заключается в следующем:

1. Предложен метод преобразования GL-моделей путем введения внутренних ребер с учетом парных реберных циклов.
2. Выполнена оценка минимального количества дополнительных внутренних ребер, необходимых для блокировки потери связности графа GL-модели с попарными реберными циклами.

**Практическая ценность** полученных в работе результатов заключается в том, что предложенный методы позволяют создавать GL-

модели, которые корректно отображают поведение систем в случае образования попарных реберных циклов, а также упрощает построение таких моделей.

**Апробация работы.** Основные положения и результаты работы были представлены и обсуждались на научной конференции магистрантов и аспирантов «Прикладная математика и компьютеринг» ПМК-2018 (Киев, 21-23 марта 2018) и международной конференции «Системный анализ и информационные технологии» SAIT 2018.

**Структура и объем работы.** Магистерская диссертация состоит из введения, трех глав и выводов.

*Во введении* представлена общая характеристика работы, выполнена оценка современного состояния проблемы, обоснована актуальность направления исследований, сформулированы цели и задачи исследований, показано научную новизну полученных результатов и практическую ценность работы, приведены сведения об апробации результатов и их внедрение.

*В первой главе* рассмотрены существующие методы построения моделей отказоустойчивых многопроцессорных вычислительных систем, а также проведен анализ, который позволяет определить основные преимущества и недостатки этих методов.

*Во втором разделе* приведены результаты исследования метода преобразования GL-моделей путем введения внутренних ребер и явление парных реберных циклов в наборе заблокированных множеств.

*В третьем разделе* формулируется метод преобразования GL-моделей путем введения внутренних ребер с учетом парных реберных циклов и предоставляется алгоритм на его основе.

*В выводах* представлены результаты проведенной работы.

*В приложениях* приведены схемы преобразования базовых GL-моделей в небазовые путем введения внутренних ребер, схемы построения V-графа модели и типы парных реберных циклов.

Работа представлена на 81 странице, содержит 6 приложений и ссылки на список использованных литературных источников из 19 наименований. В работе приведены 50 рисунков и 9 таблиц.

**Ключевые слова:** отказоустойчивые многопроцессорные системы, надежность систем, математическое моделирование, графо-логические модели.

## ЗМІСТ

СПИСОК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ МАГІСТЕРСЬКОЇ РОБОТИ.....	7
1.1 Способи моделювання відмовостійких багатопроцесорних систем.....	10
1.1.1 K-out-of-N моделі.....	10
1.1.2 Послідовні k-out-of-N моделі .....	13
1.1.3 Модель ВБС, заснована на однополюсному графі.....	15
1.1.4 Модель ВБС, що заснована на багатопольсному графі .....	21
1.2 Особливості та переваги GL-моделей.....	22
Висновки розділу .....	27
2. МЕТОДИ ПЕРЕТВОРЕННЯ БАЗОВИХ GL- МОДЕЛЕЙ.....	28
2.1 Основні поняття.....	28
2.1.1 Базові та небазові GL-моделі.....	22
2.1.2 Відображення підвищення та зниження надійності відмовостійких багатопроцесорних систем.....	33
2.2 Перетворення шляхом зміни реберних функцій.....	34
2.3 Перетворення шляхом введення додаткових ребер.....	41
Висновки розділу 2.....	58
3. МЕТОД ПЕРЕТВОРЕННЯ GL-МОДЕЛЕЙ ШЛЯХОМ ВВЕДЕННЯ ВНУТРІШНІХ РЕБЕР З УРАХУВАННЯМ ПОПАРНИХ РЕБЕРНИХ ЦИКЛІВ.....	59
3.1 Попарні реберні цикли.....	59
3.2 Відображення підвищення надійності відмовостійких багатопроцесорних систем на GL-моделях з утворенням попарних реберних циклів.....	64
Висновки розділу 3.....	76
ВИСНОВОК.....	77

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	79
-------------------------------------	----

## ДОДАТКИ

### Додаток 1. Копії графічних матеріалів

Плакат 1. Стабільні  $S$ -підмножини

Плакат 2. Побудова  $V$ -графу

Плакат 3. Побудова модифікованого графу

Плакат 4.  $V'$ -граф для моделі з попарними реберними циклами

Плакат 5. Конфігурації попарних реберних циклів

Плакат 6. Блокування попарних реберних циклів в  $GL$ -моделі

## **СПИСОК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

ВБС –відмовостійка багатопроцесорна система;

VLSI (very-large-scale integration) - інтеграція надвисокого рівня;

GL-модель – графо-логічна модель;

ПРЦ – попарні реберні цикли.

## ВСТУП

Сучасні багатопроцесорні системи складаються з більше ніж декількох тисяч різноманітних модулів. Виготовлення прототипів таких пристроїв є дуже трудомістким та дорогим процесом, оскільки для виготовлення будь-яких схем VLSI потребується виготовлення спеціальних фотолітографічних форм, вартість однієї з яких може досягати декількох сотень тисяч доларів. Тому для проектування та тестування багатопроцесорних систем сьогодні найчастіше використовують різноманітні моделі таких систем. За допомогою цих моделей можна визначити можливі стани системи модулів системи, аналізувати їх взаємодію, а також проаналізувати поведінку системи в цілому.

Одним з головних процесів, які повинні відображувати моделі відмовостійких багатопроцесорних систем, є реакція системи на відмову її модулів. Це необхідно для визначення рівня надійності шляхом проведення статистичних експериментів над моделлю. Оскільки впродовж процесу проектування ВБС такі статистичні експерименти зазвичай проводяться неодноразово, важливою характеристикою таких моделей є можливість швидко проаналізувати стан системи, що моделюється, на основі стану її модулів. Окрім того, навіть незначні зміни до структури багатопроцесорної відмовостійкої системи у процесі розробки можуть привести до значної зміни реакції системи на відмову її модулів, тому також є важливою можливістю моделей ВБС швидко змінюватись, для того щоб відповідати актуальній структурі системи.

У поданій роботі розглядається один з видів моделей ВБС, а саме GL-моделі. У роботі аналізується метод перетворення базових GL-моделей шляхом додавання внутрішніх ребер та пропонується метод, що дозволяє спростити процес визначення ребер, які необхідні для побудови небазових GL-моделей з небазових у випадку, коли множини векторів стану, які необхідно заблокувати, утворюють попарні реберні цикли.

Таким чином, GL-моделі, побудовані з використанням запропонованого методу є більш придатними для використання у розробці відмовостійких багатопроцесорних систем, оскільки метод дозволяє додавати мінімальну кількість ребер для перетворення моделі у небазову, що дозволяє пришвидшити

визначення стану системи на основі стану модулів та пришвидшує сам процес перетворення.



# **1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ МАГІСТЕРСЬКОЇ РОБОТИ**

Широке поширення обчислювальних систем та їх галузей використання, особливо для керування реальними об'єктами, суттєво вплинуло на вимоги до надійності функціонування таких систем. Сьогодні відмови та збої у роботі обчислювальних систем можуть призвести до непоправних наслідків, оскільки помилкові результати обчислень або відсутність результатів від системи керування такими складними системами, як, наприклад літальні апарати чи космічні апарати (де технічне обслуговування під час роботи є складним або неможливим), найчастіше призводить до аварій та ставить під загрозу життя людей[3]. Підвищення надійності окремих модулів таких систем чи елементної бази, з яких вони складаються, хоча і відіграє значну роль у вирішенні проблеми безвідмовної роботи систем керування, але не може цілком її вирішити. Тому для вирішення цієї проблеми широко використовуються системи, що проектуються з розрахунком на можливу відмову модулів та можуть змінювати свою конфігурацію для забезпечення виконання своїх функцій у повній мірі за відмови деяких з них. Такі системи називаються відмовостійкими обчислювальними системами.

Термін «надійність» за визначенням є властивістю об'єкта зберігати в часі в установлених межах значення всіх параметрів, що характеризують здатність виконувати необхідні функції в заданих режимах і умовах застосування, технічного обслуговування, зберігання і транспортування[1]. Відмовою системи називають подію, що призводить до її переходу у непрацездатний стан, а безвідмовність, у свою чергу, визначають як здатність безперервно зберігати працездатний стан протягом деякого часу або напрацювання.

Загалом, необхідно розрізняти вихід з ладу окремих елементів обчислювальної системи та відмову самої системи. Співвідношення понять безвідмовності та відмовостійкості визначається тим, що система залишається безвідмовною (тобто зберігає працездатність) продовж певного періоду часу як за умови, що впродовж цього часу не відмовляє жоден з її елементів, так і в тому випадку, коли відбуваються відмови тільки таких підмножин елементів, по

відношенню до яких система відмовостійка. Оскільки система у процесі експлуатації може з певною вірогідністю перейти в стан, у якому певна кількість пошкоджень не призводить до виходу з ладу самої системи (система продовжує виконувати свої функціональні обов'язки повною мірою), то можна казати про усунення наслідків несправностей[2].

Вихід системи з ладу та припинення її використання у такому випадку може бути обумовлене її переходом у граничний стан, або повною відмову системи у наслідок накопичення помилок або неможливості подальшого виконання системою своїх функціональних обов'язків повною мірою[4]. Умовами переходу системи у граничний стан можуть бути, наприклад, виникнення таких станів, за яких подальше використання системи може бути небезпечним, або станів, які розробник системи вважає індикаторами накопичення достатньої кількості помилок для того, щоб вважати систему ненадійною[5].

Для створення відмовостійких обчислювальних систем використовуються певні методи проектування, які дозволяють окрім реалізації функціональних обов'язків, необхідних згідно технічних специфікацій об'єкту, що проектується, спланувати таку обчислювальну систему, у функціональні можливості якої входить ідентифікація відмов певних елементів та підсистем, та усунення впливу цих відмов на правильність роботи головного функціоналу системи. Процес розробки такого типу зазвичай ітеративний, оскільки до початкового проекту системи, який має виконувати основний функціонал, необхідно обрати і ввести додаткові алгоритми визначення несправностей та їх усунення, алгоритми повернення системи до певного стану нормальної роботи чи її безпечного зупинення[6].

Після впровадження подібних механізмів до початкової системи, необхідно оцінити отриману надійність результуючої системи для того, щоб мати можливість порівняти її з надійністю, яку необхідно отримати згідно з технічними специфікаціями. Зазвичай така оцінка виконується методами аналітичного або електронного моделювання системи, оскільки фізичне тестування вже створеної системи може бути або неможливим, або занадто

ресурсоемним. Як правило, прогноз надійності ВБС порівнюється з прогнозом надійності багатопроцесорної системи, що не володіє стійкістю до відмов.

Ресурсна надмірність, що застосовується для таких рішень, використовується для виявлення несправності в системі і маскування виявлених помилок. У зв'язку з цим розробники обчислювальних систем мають інтерес до дослідження різноманітних методів підвищення надійності та їх впливу на процес функціонування обчислювальних систем. Для проектування подібних систем необхідно визначити, в яких випадках є сенс нарощувати надмірність обчислювальної системи та до яких меж це нарощування доцільно, який середній час напрацювання на відмову системи з обраною структурою і як найбільш ефективно використовувати запроваджену надмірність для забезпечення необхідної надійності.

У процесі створення відмовостійкої керуючої обчислювальної системи постає проблема оцінки надійності для спроектованої системи певної ітерації для порівняння її з надійністю, що потребується згідно специфікацій на кінцеву розробку. Зі збільшенням складності апаратних і програмних засобів, які виконують обробку інформації, збільшенням швидкості обробки та об'ємів інформації, постає проблема ефективної оцінки надійності системи, оскільки звичайні методи оцінки надійності або не можуть бути застосовані, або потребують значних витрат.

Процес виявлення неполадок у відмовостійкій багатопроцесорній системі виконується впродовж наступних головних етапів життя системи:

1. Етап виробництва (початковий контроль, виконується до експлуатації обчислювальної системи);
2. Етап експлуатації (оперативний контроль, виконується під час виконання системою її основних функцій);
3. Етап автономного контролю (такий контроль виконується при тимчасових перервах у роботі обчислювальної системи).

Кожен з цих етапів, як вже було зазначено, передбачує подальше збільшення кількості модулів у системі, що призводить до можливості відмови більшої їх кількості процесі роботи системи. Тому використання зовнішніх

засобів контролю и діагностування в таких системах є, зазвичай, невиправданим, оскільки втручання таких засобів у роботу системи призводить до її потенційного ускладнення і зниження продуктивності. При цьому додатково постає проблема забезпечення надійної роботи цих засобів (такі зовнішні системи теж мають певну обмежену надійність і потребують тестувань і перевірки). Тому при визначенні працездатності системи і діагностуванні несправностей в ній доцільно використовувати взаємодії модулів між собою, що виникають в процесі їх функціонування.

Враховуючі ці особливості проектування ВБС, для оптимального вибору апаратних и алгоритмічних ресурсів дуже велика необхідність отримання оцінки параметрів надійності ВБС достатньо великої точності на ранніх етапах її проектування. У той самий час відомі методи розрахунку параметрів надійності у випадку ВБС довільної складності дають неточну оцінку. Спроби наблизити розрахункові оцінки до реальних стикаються з непереборними обчислювальними труднощами, що пов'язані зі складністю перебору великої кількості можливих станів системи.

Тому, до задачі розробки адекватної математичної моделі поведінки ВБС у входить, у першу чергу, спрощення обчислення результатів тестування реакції системи на певний стан елементів, що її складають. Таким чином, збільшивши швидкість отримання результату для одного стану системи, можна збільшити кількість станів, які можуть бути оброблені за доречний час, в такий спосіб збільшивши точність статистичного аналізу параметрів надійності системи, що моделюється.

## **1.1 Способи моделювання відмовостійких багатопроцесорних систем**

Для моделювання багатопроцесорних відмовостійких систем існує достатньо широке різноманіття методів. Більшість з них фокусуються на визначенні стану системи за станом її модулів.

### **1.1.1 K-out-of-N моделі**

K-out-of-N системи – це системи з резервуванням у режимі очікування та характерним режимом обслуговування, що відображається на методах їх

моделювання. Типовим прикладом такої системи є решітки активних радарів, що складаються з великої кількості елементів.

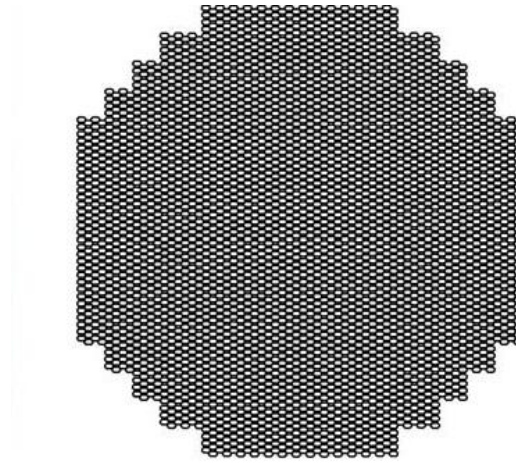


Рис. 1.1. Решітка активного радару як приклад k-out-of-N системи

Системи такого типу розглядаються у часі, визначаючи процес відмови модулів негативним експоненціальним розподілом з інтенсивністю  $\lambda$  з припущенням, що процес виходу з ладу кожного модулю не залежить від інших модулів.

Графік, що зображує ймовірність виходу з ладу модулю k-out-of-N системи приведено на рисунку 1.2.

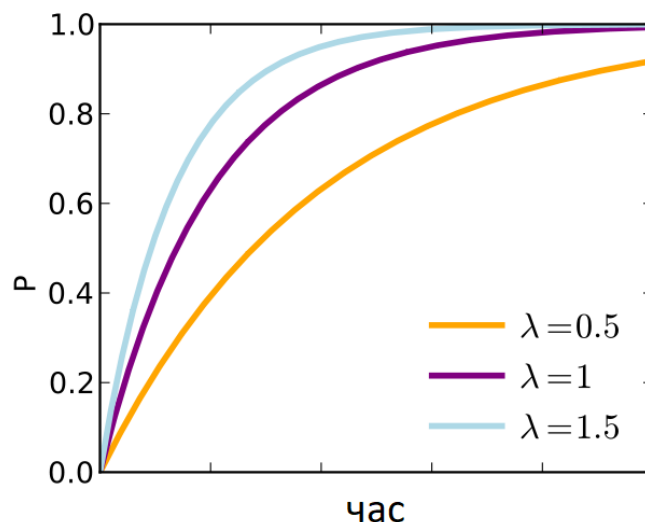


Рис. 1.2. Графік ймовірностей виходу з ладу модулів k-out-of-N системи

Система залишається роботоздатною, якщо не більше ніж  $N-k$  модулів виходять з ладу. Окрім того, щоб запобігти простою системи в разі втрати роботоздатності, при виході з ладу  $m \leq N-k$  модулів, система ініціює процедуру

технічне обслуговування, впродовж якої модулі, що вийшли з ладу, замінюються на нові. Видається доцільним обрати  $m=N-k$ , якщо витрати на заміну обладнання є високими, але менші  $m$  можуть бути обрані, якщо між ініціацією технічного обслуговування та фактичним початком робіт існує певна затримка[7].

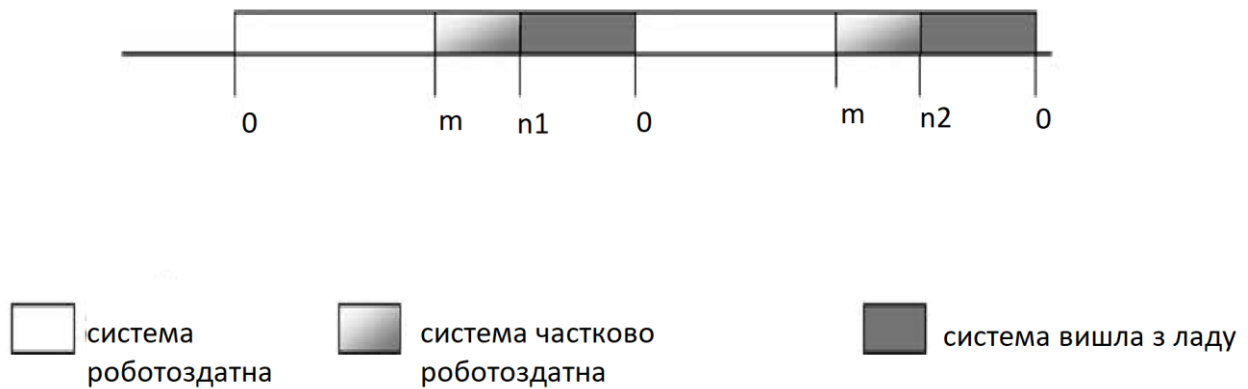


Рис. 1.3. Схема циклу функціонування k-out-of-N системи

K-out-of-N моделі зосереджуються на оцінці очікуваного часу роботоздатності системи (за умов кінцевої кількості запасних модулів та обмеженої можливості поповнення їх запасів), визначаючи його за наступною формулою:

$$AE_{m,s,c} = \frac{E(T_m) + E(U_m)}{E(T_m) + L + E(D_{m,s,c})} \quad (1.1)$$

де  $E(T_m)$  – очікуваний час до початку технічного обслуговування,  $E(U_m)$  – час, за який з початку обслуговування хоча б  $k$  модулів будуть замінені,  $L$  – затримка між ініціацією технічного обслуговування та фактичним початком робіт, а  $E(D_{m,s,c})$  – очікуваний час, необхідний для відновлення системи у новий стан.

Для нульової затримки між ініціацією технічного обслуговування та фактичним початком робіт, необхідно розрахувати лише  $E(T_m)$  та  $E(D_{m,s,c})$  (оскільки  $L$  та, відповідно,  $E(U_m)$  дорівнюють нулю). Очікуваний час  $E(T_m)$  можна розрахувати за наступною формулою:

$$E(T_m) = \sum_{i=0}^m \frac{1}{(N-i)\lambda} \quad (1.2)$$

Для розрахунку очікуваного часу  $E(D_{m,S,c})$  може бути використане наступне рівняння:

$$E(D_{m,S,c}) = \sum_{s=0}^S E[R_c(m-s, S-s+m|s)] \pi_{m,S,c}(s) \quad (1.3)$$

де  $R_c(m-s, S-s+m|s)$  – це час на відновлення  $m-s$  модулів використовуючи  $c$  постачальників коли  $S-s+m$  модулів очікують на ремонт, а  $\pi_{m,S,c}(s)$  – стабільна вірогідність наявності  $s$  запасних модулів, готових до використання на початку періоду технічного обслуговування, за відомих  $m$ ;  $S$  і  $c$  [8].

Для того щоб розрахувати очікуваний час  $E[R_c]$ , використовується наступна формула:

$$ER_c(i, j) = \begin{cases} 0; i \leq 0 \\ \sum_{h=0}^{i-1} \frac{1}{(j-h)\mu}; 0 < i \leq j \leq c \\ \frac{i}{c\mu}; j > c, i \leq j - c \\ \frac{j-c}{c\mu} + \sum_{h=0}^{i-j+c-1} \left\{ \frac{1}{(c-h)\mu} \right\}; j > c, j - c < i \leq j \end{cases} \quad (1.4)$$

Стабільна вірогідність наявності  $s$  запасних модулів  $\pi_{m,S,c}(s)$  розраховується за допомогою ланцюга Маркова [9].

K-out-of-N моделі дозволяють визначити не тільки стан системи в залежності від стану її елементів, але й робити прогнози щодо часу, який буде необхідним для технічного обслуговування таких систем за певних умов цього технічного обслуговування. Однак, для більш складних систем, в яких певні модулі є більш важливими ніж інші, та роботоздатність системи залежить не тільки від кількості модулів, що відмовили, а й від їх типу, такі моделі не підходять.

### 1.1.2 Послідовні k-out-of-N моделі

Однією з відомих модифікацій k-out-of-N моделей є так звані послідовні k-out-of-N, що відображують реакцію системи на відмову модулів, з яких вона

складається, для таких систем, де має значення взаємне положення модулів у певній структурі.

Таким чином, послідовні k-out-of-N системи втрачають можливість функціонувати, коли послідовність з k модулів такої системи виходить з ладу. Якщо певні модулі загальною кількістю k відмовили, але не знаходяться один за одним, системи вважається роботоздатною[10].

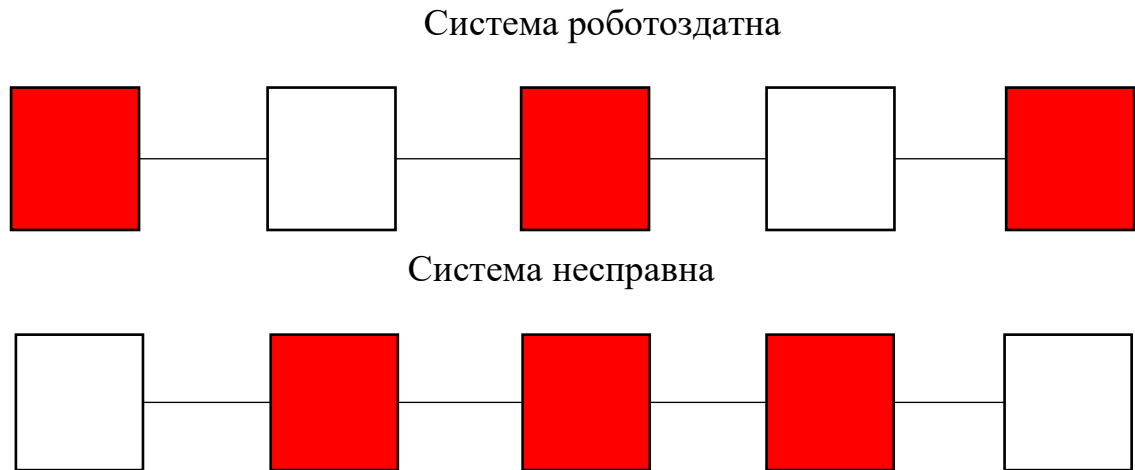


Рис. 1.4. Залежність стану послідовної k-out-of-N системи від взаємного положення модулів, що відмовили

Для лінійних послідовних k-out-of-N відмовостійкість визначається за наступною формулою:

$$R(n, k) = R(n - 1, k) + Q(n - k - 1, k)u_{n-k} \prod_{i=n-k+1}^n r_i \quad (1.5)$$

де:

- $Q(n, k) = 1 - R(n, k)$
- $r_i$ - відмовостійкість i-елементу;
- $u_i = 1 - r_i$

У випадку, коли відмовостійкість кожного елементу системи є однаковою, тобто:

$$r_1 = r_2 = \dots = r_n \quad (1.6)$$

обчислити відмовостійкість можна за наступним рівнянням:

$$R(n, k) = R(n - 1, k) + Qn r^k [1 - R(n - k - 1, k)] \quad (1.7)$$



Такі системи спрощують застосування  $k$ -out-of- $N$  моделей у багатьох галузях. Наприклад, такі моделі використовують для оцінки відмовостійкості телекомунікаційних систем, кільцевих комп'ютерних систем, космічних ретрансляційних станцій, тощо.

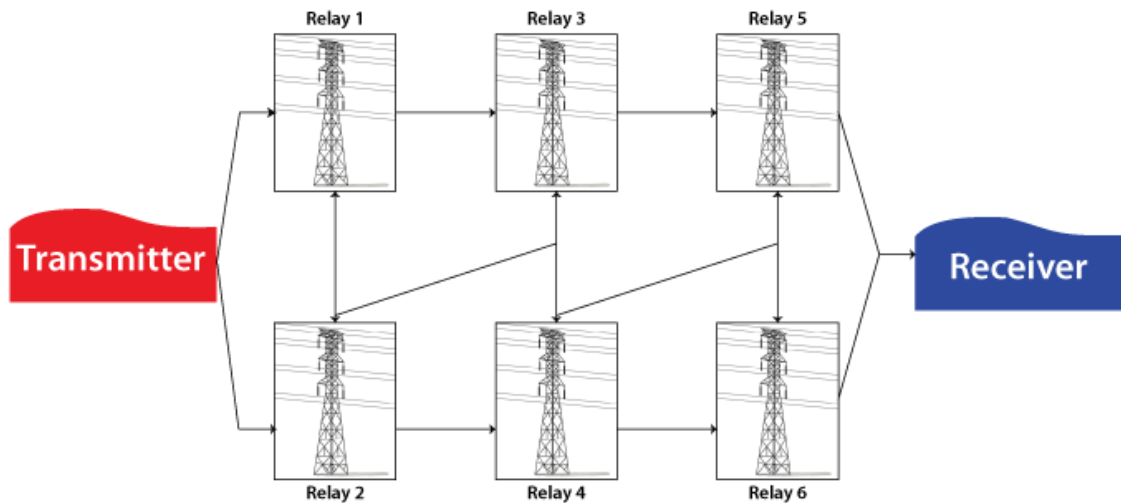


Рис. 1.5. Ретрансляційні станції як приклад послідовних  $k$ -out-of- $N$  систем

Однак для складних систем типу ВБС неможливість відобразити більш складні взаємні залежності між модулями та їх вплив на стан системи під час відмови модулів, обмежує використання моделей такого типу[11].

### 1.1.3. Модель ВБС, заснована на однополюсному графі

У сучасних моделях широко застосовується низка підходів, які інтерпретують задачу розрахунку показників надійності ВБС як задачу розрахунку певних характеристик графу (зазвичай зв'язності).

Графом називають впорядковану сукупність двох множин  $V$  та  $E$ :  $V$  – множина вершин (вузлів) графу, а  $E$  – множина пар вершин з  $V$ , які називають ребрами. Ребра графу можуть бути неорієнтованими або орієнтованими (у цьому разі їх називають дугами). Якщо ребро зв'язує дві вершини, то говорять, що воно інцидентне цим вершинам. Два ребра є суміжними, якщо вони інцидентні одній і тій ж вершині. У випадку орієнтованого графу, дуга  $\{u, v\}$  інцидентна до вершин  $u$  і  $v$ , при цьому говорять, що  $u$  — початкова вершина дуги, а  $v$  — кінцева вершина. При цьому наявність дуги  $\{u, v\}$  гарантує, що існує шлях з  $u$  до  $v$ , але не гарантує, що є шлях з  $v$  до  $u$ .

Якщо в кінцевому графі (тобто в графі, що включає кінцеве число вершин), з будь-якої вершини через кінцеве число ребер (або дуг) можна потрапити в будь-яку іншу вершину, то такий граф називається зв'язним.

Модель ВБС, заснована на двополюсному графі зводить задачу розрахунку показників надійності системи до задачі визначення зв'язності двополюсного графу (саме такий вигляд мають схеми розрахунку надійності: у них є два полюси - вхідний і вихідний). Це можна інтерпретувати на фізичному рівні розрахунок можливості надходження деякого сигналу від входу системи до виходу.

Першим, що необхідно зробити за для аналізу технічної системи, це визначити критерій відмови. Критерієм відмови реальної системи, зазвичай, буде подальша неможливість системи виконувати свої основні функції. Математична модель, в більшості випадків, такі критерії визначаються через стани елементів, з яких складається технічна система, що моделюється. У такому випадку кожен з елементів, що входить до складу системи, вважається найпростішим, тобто його стан визначається бінарною функцією: або елемент працездатний, або ні. Відповідно, вектор стану системи визначається як сукупність станів усіх елементів, з яких вона складається.

Для позначення стану  $i$ -го елемента системи вводиться індикаторна функція:

$$x_i(t) = \begin{cases} 1, \text{ якщо } i - \text{й елемент працездатний} \\ 0, \text{ якщо } i - \text{й елемент відмовив} \end{cases} \quad (1.8)$$

Моделі, які розглянуті у поданій роботі, вважають що стан системи зафіксовано у певний момент часу. У такому разі можна замість індикаторної функції використовувати множини значень цих функцій у цей момент, отримавши звичайні булеві змінні. Якщо позначити ці змінні як  $x_1, x_2, \dots, x_n$  для моделі системи, що складається з  $n$  елементів, то вектор  $X = (x_1, x_2, \dots, x_n)$  буде характеризувати стан системи [12,13].

Розглянувши стан системи у поданому вигляді, можна дійти висновку, що безліч можливих станів системи складається з  $2n$  різних станів. Кожен з цих станів можна віднести до однієї з двох підмножин: підмножина станів, у якій система є працездатною, та підмножина, у якій система є непрацездатною. Це

можна подати як задану на множині  $X$  функцію  $F(X)$  (яка у математичній теорії надійності називається структурною) таку, що:

$$F(X) = \begin{cases} 1, \text{ якщо стан } X \text{ відповідає працездатному стану системи} \\ 0, \text{ якщо стан } X \text{ відповідає непрацездатному стану системи} \end{cases} \quad (1.9)$$

Треба підкреслити, що хоча б два стани системи вважають завжди однозначно детермінованими. Стан  $x_i = 1, i=\overline{1, n}$ , коли кожен з елементів функціонує, вважають станом, коли система апріорно функціонує. Натомість, стан  $x_i = 0, i=\overline{1, n}$ , коли кожен з елементів відмовив, вважають таким, коли вся система не функціонує. Для усіх інших станів значення  $F(X)$  підбирається згідно з характеристиками та вимогам до функціонування реальної системи, що моделюється.

Моделі ВБС, які засновані на двополюсному графі базуються на наступних базових системах[13]:

#### 1. Послідовна система

Послідовною системою називають систему, відмова будь-якого з елементів якої призводить до непрацездатного стану всієї системи. У вигляді графу така система виглядає наступним чином:

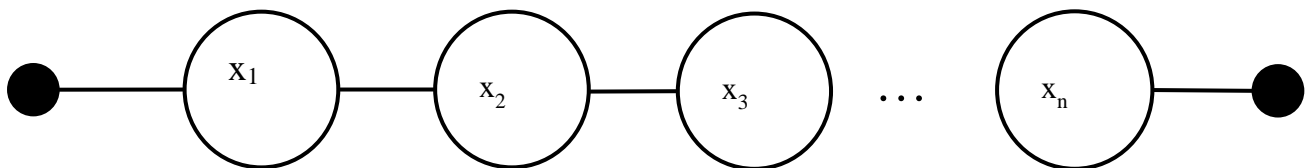


Рис. 1.6. Схема послідовної системи

Структурна функція для такої системи буде наступною:

$$F(X) = \bigwedge_{i=1}^n x_i \quad (1.10)$$

Якщо у наявності не лише дані щодо виходу системи з ладу за певного стану елементів, з яких вона складається, то можна розрахувати надійність такої системи. Якщо позначити вірогідність безвідмовної роботи  $i$ -го елемента як  $p_i$ , то надійність усієї системи можна розрахувати за наступною формулою:

$$P(F(X)) = \prod_{i=1}^n p_i \quad (1.11)$$

## 2. Паралельна система

Паралельною є будь-яка система, працездатність хоча б одного з елементів якої забезпечує працездатність усієї системи.

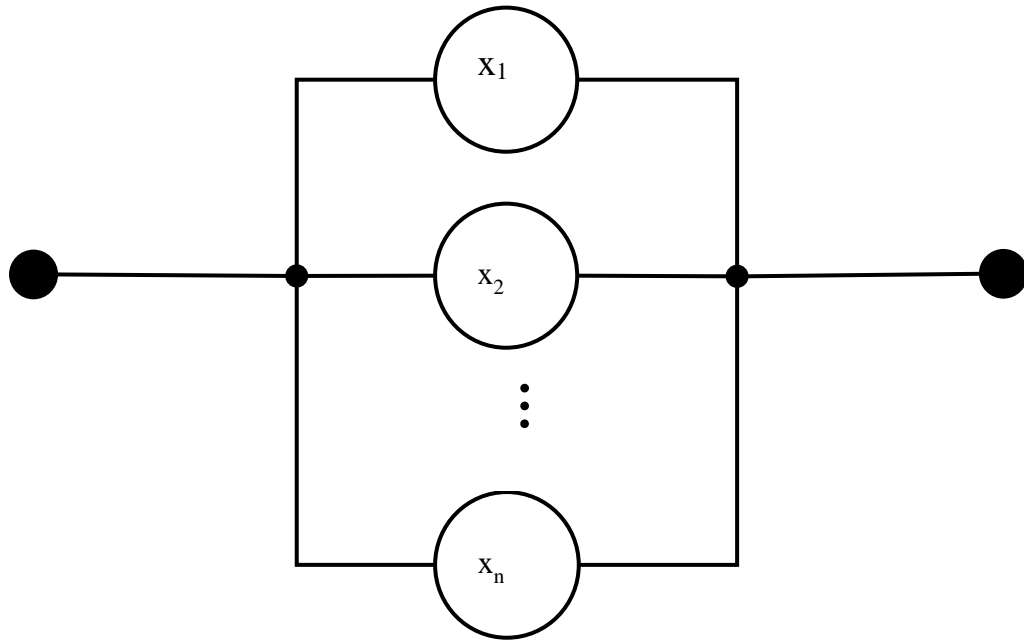


Рис. 1.7. Схема паралельної системи

Структурна функція такої системи буде мати наступний вигляд:

$$F(X) = \bigvee_{i=1}^n x_i \quad (1.12)$$

Якщо позначити вірогідність безвідмовної роботи  $i$ -го елементу як  $p_i$ , то надійність усієї системи можна розрахувати за наступною формулою:

$$P(F(X)) = \prod_{i=1}^n (1 - p_i) \quad (1.13)$$

## 3. Послідовно-паралельна система

Така система являє собою сукупність послідовних підсистем, з'єднаних паралельно (рис. 1.8).

Структурна функція такої системи буде мати вигляд:

$$F(X) = \bigvee_{j=1}^k \bigwedge_{i=1}^{n_j} x_{ji} \quad (1.14)$$

У вигляді графа подана система буде виглядати наступним чином:

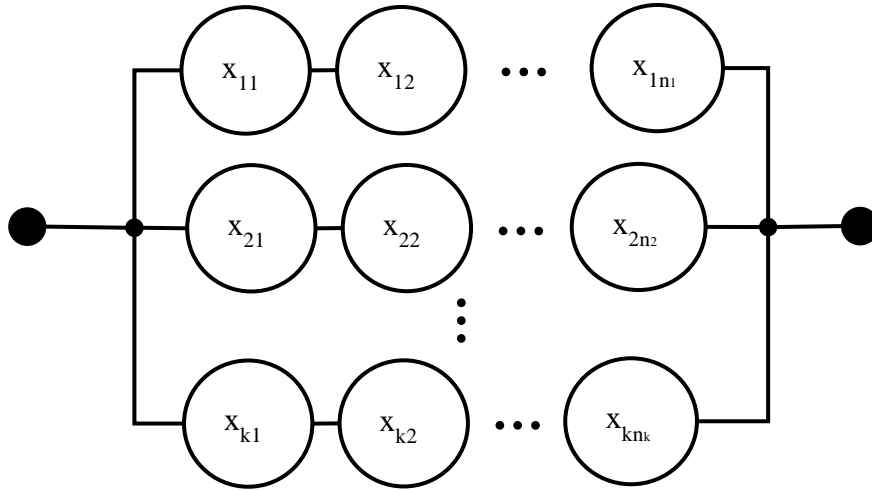


Рис. 1.8. Схема послідовно-паралельна системи

Якщо  $p_{ji}$  – це вірогідність безвідмовної роботи елементу  $x_{ji}$ , то надійність системи буде дорівнювати:

$$P(F(X)) = 1 - \prod_{j=1}^k (1 - \prod_{i=1}^{n_j} p_{ji}) \quad (1.15)$$

#### 4. Паралельно-послідовна система

Ця система являє собою сукупність паралельних підсистем, з'єднаних паралельно.

Структурна функція такої системи має вигляд:

$$F(X) = \bigwedge_{j=1}^k \bigvee_{i=1}^{n_j} x_{ji} \quad (1.16)$$

У вигляді графу подана система буде виглядати наступним чином:

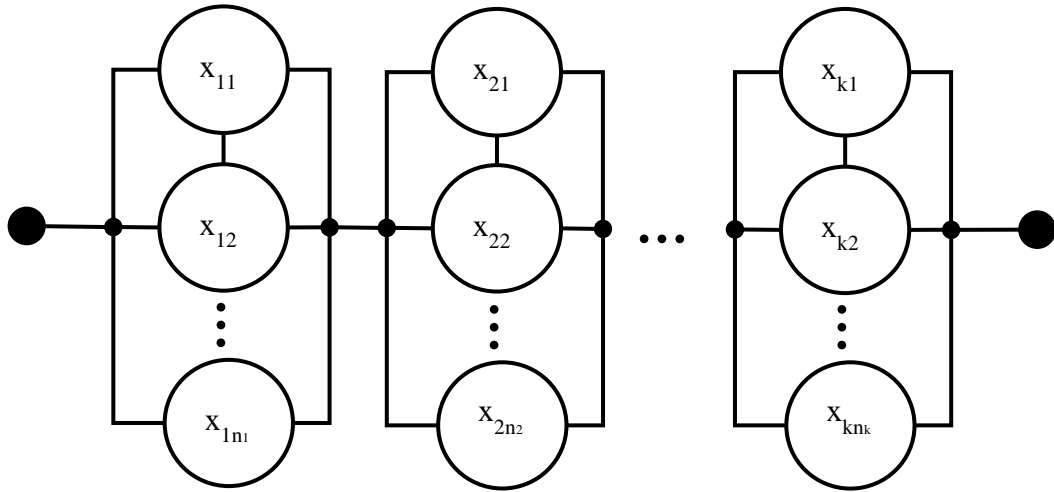


Рис. 1.9. Схема паралельно- послідовної системи

Якщо  $p_{ji}$  – це вірогідність безвідмовної роботи елементу  $x_{ji}$ , то надійність системи буде дорівнювати:

$$P(F(X)) = \prod_{j=1}^k (1 - \prod_{i=1}^{n_j} (1 - p_{ji})) \quad (1.17)$$

Окрім того, виділяють системи зі структурою, що приводиться, або не приводиться. До систем структурою, що приводиться, відносяться такі системи, які шляхом заміни послідовних чи паралельних підсистем деякими еквівалентними підсистемами можуть бути зведені до найпростішої двополусної системи, тобто до абстрактного елементу. Якщо система за допомогою таких еквівалентних перетворень не може бути зведена до елементу, то структура такої системи не приводиться. Зі збільшенням кількості ребер у моделі системи, частка наведених структур у загальному обсязі всіх можливих структур суттєво зменшується. У той же час не існує структур, які відображаються графом з чотирма і менш ребрами і одночасно не можуть бути зведені до елементу. Якщо кількість ребер дорівнює п'яти, то існує одна структура, яка не приводиться - так звана схема мосту.

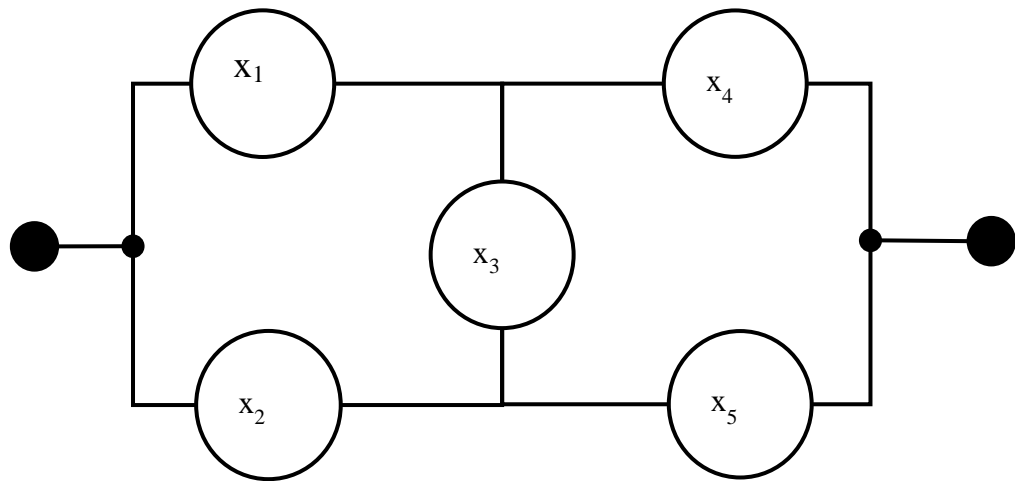


Рис. 1.10. Схема мосту

Істотним недоліком структур, що не приводяться, є складність побудови структурних функцій. Якщо кількість елементів в системі невелика, то можна скласти таблицю істинності або використовувати розкладання на простіші функції. Таке розкладання можна виконати, наприклад, за допомогою розкладання(декомпозиції) Шеннона по змінній  $x_i$  – методу представлення булевої функції багатьох змінних, у вигляді суми(диз'юнкції) двох підфункцій від змінних, що залишилися(усіх, окрім  $x_i$ ). Розділення функцій базується на тому, що таблицю істинності булевої функції від  $n$  змінних можна розбити на дві частини таким чином, що до першої будуть входити усі вхідні комбінації, за яких  $x_i$  дорівнює 0, а до другої – усі вхідні комбінації, за яких  $x_i$  дорівнює 1. У результаті стає справедливим наступне тотожність, називають розкладанням Шеннона[14]:

$$F(X) = x_i F(X_{x_i=1}) \vee \bar{x}_i F(X_{x_i=0}) \quad (1.18)$$

У відносно складних графах ефективний вибір ребра, щодо якого слід проводити розкладання структурної функції, представляє складну і невирішену на даний момент завдання. Для таких графів при розрахунку надійності використовуються методи, в результаті яких після перерахування всіх мінімальних шляхів і мінімальних розрізів можна отримати оцінки надійності. Головним недоліком цього методу є те, що рішення виходить приблизне, у вигляді двох оцінок: оцінки знизу і оцінки зверху, що найчастіше обмежує використання цього методу на практиці.

### 1.1.4 Модель ВБС, що заснована на багатополіусному графі

За необхідності оцінки працездатності систем, що мають складну розгалужену мережеву структуру, дослідження зв'язності двох полюсів (вирішення задачі, чи може певний вузол (джерело) здійснити зв'язок з певним іншим вузлом (стоком) не може дати вичерпної відповіді критерію щодо якості функціонування системи. Для випадку ВБС надійність досить часто обчислюється за критерієм повнозв'язності графа.

У випадку використання такого методу для опису системи використовується неорграф. Оскільки критерієм працездатності системи при певному стані елементів є повнозв'язність графа, то задача визначення стану системи зводиться до визначення чи можна з кожного вузлу графа досягти будь-якого іншого вузла. Це відповідає випадку, коли хоча б одне з остових дерев неографа є працездатним.

При використанні в якості моделі для розрахунку надійності багатополюсного графа комбінаторна задача перебору всіх остових дерев для складних систем призводить, як і в випадку з двополюсними графами, до значних обчислювальних труднощів. Графо-логічна модель ВБС дозволяє зробити аналіз системи більш доступним і доступним для огляду, в результаті чого буде спрощено розрахунок параметрів надійності і отримати досить точну оцінку цих параметрів[15].

## **1.2 Особливості та переваги GL-моделей**

ВБС, що складається з  $n$  модулів, стійка до відмов певної кратності (тобто стійка до  $m$  відмов), називається базовою ВБС і позначається  $m$ -ВБС, а її модель –  $K(m, n)$ . Однак такі системи далеко не завжди відображають реальні можливості побудови ВБС. Реальна відмовостійка система управління складним об'єктом зазвичай має унікальний, нерегулярний набір векторів стану модулів, на яких вона залишається працездатною. Окрім того, кількість компонент, які визначають працездатність реальної системи занадто велике, і може вимірюватись сотнями і тисячами[16].

Тому актуальності набуває проблема створення нових моделей поведінки ВБС, які б дозволили зробити аналіз станів системи більш простим та менш ресурсоємним, давши змогу обчислювати більшу кількість станів у певну



одиницю часу  $i$ , таким чином, збільшити точність статистичних розрахунків надійності. Існуючі моделі бездоганно виконують свої задачі для простих систем, але зі збільшенням складності вони, зазвичай, або зовсім не можуть бути застосовані, або потребують великої кількості складних обчислень, що робить їх неефективними.

Для подолання цих перешкод створено відносно новий тип моделей, які поєднують у собі використання властивостей графів та булевих функцій – так званих графо-логічних моделей ВБС (GL-моделей).

Моделі такого типу передбачають використання зв'язності певного графу як критерію працездатності системи, однак на відміну від розглянутих вище моделей, які також використовують граф у якості інструменту для опису системи, GL-моделі мають певні особливості. У поданому випадку граф може мати більш просту структуру, оскільки частина складності взаємодії елементів системи «переноситься» на булеві функції, що асоціюються з ребрами графу.

Головна мета такого підходу полягає у тому, що, обравши регулярну форму представлення булевих функцій у графо-логічній моделі, можна змінювати структуру графу у залежності від вектору стану елементів системи у залежності, що досягається шляхом зв'язування ребер графу з певними булевими функціями.

Якщо поставити у відповідність  $m$ -ВБС певний зв'язний неорієнтований граф  $G$ , ребра якого позначаються певними булевими функціями  $f_1(x_1, \dots, x_n), \dots, f_L(x_1, \dots, x_n)$ , де кожна змінна  $x_i$  відображає стан  $i$ -го елементу системи аналогічно розглянутим вище моделям ( $x_i = 1$ , якщо  $i$ -й елемент системи працездатний, та  $x_i = 0$  в іншому випадку), то ребро, яке позначене функцією  $f_i(x_1, \dots, x_n)$ , залишається у графі, якщо  $f_i(x_1, \dots, x_n) = 1$  та видаляється з нього, якщо  $f_i(x_1, \dots, x_n) = 0$ .

Обравши реберні функції графу таким чином, щоб граф  $G$  втрачав зв'язність тільки у тих випадках, коли не менше  $m+1$  з  $n$  змінних дорівнювали 0, можна відмову  $n$ -модульної системи можна інтерпретувати, як втрату зв'язності графа  $G$  (граф називається зв'язним, якщо будь-яка пара вершин є пов'язаною, тобто якщо між ними існує маршрут).

У загальному випадку (тобто якщо ВБС не є базовою) необхідно забезпечити втрату зв'язності графа  $G$  при появі будь-якого вектора стану, який відображає відмову ВБС, і збереження зв'язності при появі всіх інших векторів стану. Зауважимо, що відмови елементів ВБС не завжди можуть приводити до зникнення ребер у графі моделі; втрата працездатності самої ВБС означає втрату таких ребер в графі, які призводять до втрати зв'язності.

Під час створення GL-моделі постійно постають питання обрання графу  $G$  та функцій, що відповідають його ребрам. Існує багато варіантів GL-моделей одної й тієї ж самої ВБС, яка використовує різні підходи при розподіленні складності між розгалуженням графу та обчислювальною складністю реберних функцій. У цій роботі буде розглядатись одна конфігурація графу, яка найбільш часто використовується – циклічна.

Графи моделі циклічного типу являють собою замкнений ланцюг з ребер, кількість ребер та правила створення відповідних їм функцій добре описані у багатьох наукових працях. Використання моделей подібного типу для створення  $K(m, n)$  моделей ВБС базується на тому, що для втрати зв'язності такого графу необхідно, щоб граф втратив 2 чи більше ребер. У рамках GL-моделей це відповідає такому значенню вектору стану, за якого хоча б 2 з  $f_1(x_1, \dots, x_n), \dots, f_L(x_1, \dots, x_n)$  дорівнюють 0.

Наприклад, GL-модель для  $K(2, 6)$  системи буде мати наступний вигляд:

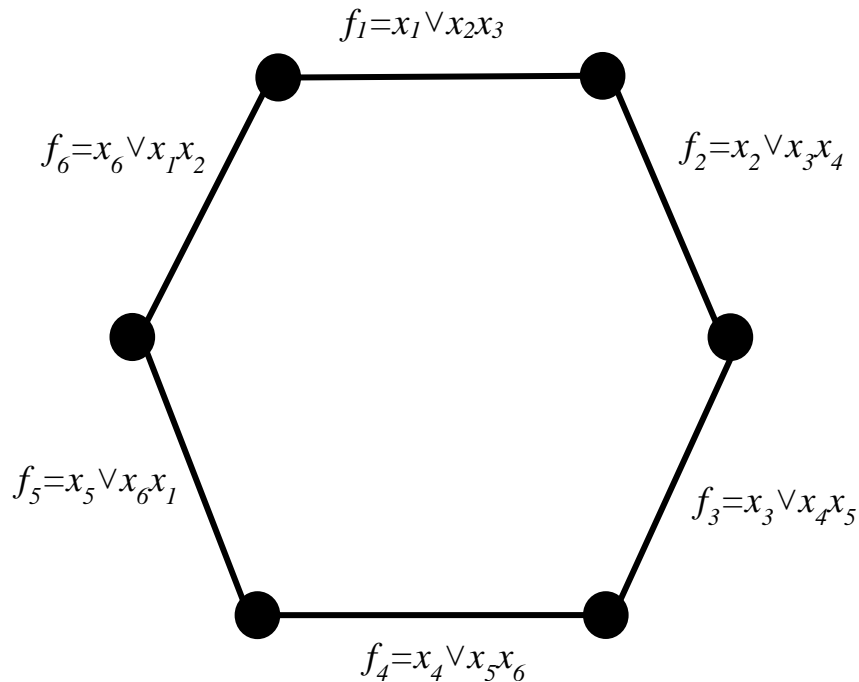


Рис. 1.11. Схема GL-моделі  $K(2, 6)$  системи

Незважаючи на те, що методологія створення базових GL-моделей на основі замкнених ланцюгів добре досліджена, проблема створення з них небазових моделей довільних систем залишається відкритою. Існуючі підходи або потребують складних рекурсивних обчислень, або працюють для обмеженої кількості моделей.

Більшість небазових GL-моделей зображують системи, у яких окрім стійкості до відмови довільних  $m$  елементів, існує також стійкість до певних комбінацій з  $m+1$  елементів. Побудова таких моделей потребує досить важких обчислень у порівнянні з побудовою базових моделей. Окрім того, складність створення подібних моделей великою мірою залежить від кількості таких наборів з  $m+1$  елементів необхідно блокувати (змінити модель таким чином, щоб відповідний стан елементів відображувався моделлю як працездатний).

Зі збільшенням кількості наборів елементів, які необхідно блокувати, зростає складність створення небазових моделей з базових. Якщо кількість таких наборів сягне половини можливих комбінацій з  $n$  по  $m+1$  елементів, то здається доречним використання зворотного методу – модифікувати базову  $K(m+1, n)$  GL-модель ВБС таким чином, що усі комбінації з  $m+1$  елементів, які

не входять у список тих, за яких система продовжує працювати, мають бути розблоковані (функції моделі мають бути змінені таким чином, щоб відповідний стан елементів відображувався моделлю як непрацездатний). Такий підхід має бути тим більш ефективним, чим більша кількість елементів потребували б блокування у класичному підході.

## **Висновки розділу 1**

В даному розділі проведено аналіз існуючих методів моделювання відмовостійких багатопроцесорних систем. При цьому показано специфіку можливостей кожного типу моделей, проаналізовані характерні вади кожного з методів. Однак, було встановлено, що для моделювання систем з залежностями стану системи від типу модулів, що відмовили, необхідні значні модифікації вже існуючих моделей, які ускладнюють структуру моделі та збільшують час, необхідний для визначення стану системи та час, необхідний для створення модифікованої моделі. Все це обґрунтовує необхідність нових, ефективніших методів перетворення моделей для відображення реакції ВБС на відмову модулів.

## 2. МЕТОДИ ПЕРЕТВОРЕННЯ БАЗОВИХ GL-МОДЕЛЕЙ

### 2.1 Основні поняття

#### 2.1.1 Базові та небазові GL-моделі

GL-моделі поєднують у собі використання властивостей графів та булевих функцій для більш компактного відображення критерію, у відповідності з яким виконується перевірка поведінки системи в умовах, що приймаються за певний стан об'єкту, що моделюється.

На відміну від інших моделей, що використовують графи, GL-моделі пропонують використання графів з менш складною структурою. Це досягається шляхом відображення поведінки системи на зв'язність графа і функціями, що обчислюються для кожного з станів та ставляться у відповідність кожному ребру цього графа. Ці функції, обрані певним визначеним чином, дозволяють перенести обчислювальну складність зі структури графу на складність булевих виразів, з яких вони складаються.

При цьому критерій, за яким визначається стан системи у цілому може бути досить тривіальним, якщо є можливість обрати регулярну форму представлення булевих функцій у GL-моделі.

Головна задача під час створення GL-моделі полягає у тому, щоб обрати базову структуру графу, яка буде використовуватись у моделі, та підібрати функції, що будуть відповідати його ребрам таким чином, щоб створення базової моделі не потребувало великої кількості обчислювальних операцій, а подальша модифікація моделі не потребувала складних трансформацій функцій, які поставлено у відповідність ребрам графу моделі, чи повної перебудови структури обраного графу.

Для початку розглянемо найпростіший варіант, за яким система, що складається з  $n$  модулів та стає непрацездатною за умови відмови не менш ніж  $m+1$  модулів, де  $m < n$ . Такі ВБС зазвичай у літературі називають  $m$ -ВБС.

Як вже було зазначено вище, у відповідність  $k$ -ВБС необхідно поставити деякий неорієнтований граф  $G$ , ребра якого позначаються булевими функціями  $f_1(x_1, \dots, x_n), \dots, f_L(x_1, \dots, x_n)$ , де кожна змінна  $x_i = x_1 \dots x_n$  є індикативною функцією, що позначає стан  $i$ -го елемента системи, тобто  $x_i = 1$ , якщо  $i$ -й елемент системи

працездатний, та  $x_i = 0$  інакше. Вважається, що ребро, яке позначено функцією  $f_i(x_1, \dots, x_n)$  залишається у графі  $G$ , якщо  $f_i(x_1, \dots, x_n) = 1$ , і є видаленим, якщо  $f_i(x_1, \dots, x_n) = 0$ .

У більшості моделей функції  $f_1(x_1, \dots, x_n), \dots, f_L(x_1, \dots, x_n)$  обирають таким чином, щоб граф  $G$  втрачав зв'язність тільки у тому випадку, коли нулю дорівнюють не менш  $m + 1$  будь-яких з  $n$  змінних, то відмову  $n$ -модульної системи можна було відобразити втратою зв'язності графа  $G$ .

Це відповідає умові, що функції  $f_1(x_1, \dots, x_n), \dots, f_L(x_1, \dots, x_n)$  для будь-якого вектору стану, що містить будь-яке поєднання з не менше ніж  $k+1$  нульових розрядів, містять  $r$  ( $r < L$ ) функцій, які приймають нульові значення на цих наборах і позначають  $r$  таких ребер, видалення яких з графа  $G$  призводить до втрати його зв'язності.

Найпростішим варіантом графу  $G$  є незамкнений ланцюг, а найпростішими функціями –  $f_i = x_i$ . Модель, створена таким чином, відповідає графо-логічній моделі для 0-ВБС. Функції для створення  $m$ -ВБС на базі такого графу мають мати наступну властивість: хоча б одна з функцій для всіх станів, який містить у собі  $k$  та більше модулів, що відмовили, має дорівнювати 0, тим самим руйнуючи зв'язність на цих наборах [17].

Більш перспективною конфігурацією для організації моделі є циклічний граф. Якщо отримати такий циклічний граф шляхом замикання ланцюгової моделі, що була описана вище, то за тими ж критеріями інтерпретації відмови системи можна отримати 1-ВБС, не збільшуючи при цьому обчислювальну складність системи.

На основі замкненого циклічного графу можна створювати моделі системи будь-якої складності. Для прикладу можна розглянути випадок, коли система, що складається з  $n$ -модулів, стає непрацездатною при відмові трьох і більше модулів, тобто 2-ВБС. Відомо наступний метод: використовуючи у якості графу замкнений циклічний граф з  $n$  ребер, кожному з яких відповідає певна функція:

$$f_i(x_1, \dots, x_n) = \psi_i \vee \varphi_i, i = \overline{1, n} \quad (2.1)$$

де:

$$\psi_i = \bigwedge_{j=1}^{\left\lfloor \frac{n-1}{2} \right\rfloor} x_{(i+j) \bmod(n)} \quad (2.2)$$

$$\varphi_i = x_i$$

Покажемо, що модель, яка створена за цим алгоритмом, відповідає поведінці 2-ВБС. Функції графа не містять логічної функції «ні», тому зв'язність графу не буде порушена для векторів, що відображують стани, у яких відмовив лише один модуль, якщо модель системи є стійкою до будь-яких векторів, що містять у собі 2 змінні, які дорівнюють логічному нулю. Виходячи з цього, достатньо показати стійкість моделі до таких векторів.

Окрім цього, необхідно довести, що зв'язність буде порушуватись для будь-яких векторів, що містять 3 змінні, які інтерпретуються як стан відмови відповідного модуля. Відсутність необхідності показувати такі ж закономірності для кількості модулів, що відмовили, більшої за 3 також впливає з відсутності інверсій у формулах.

Припустимо, що два довільних модулі з номерами  $p$  та  $q$  вийшли з ладу. У такому разі  $x_p$  та  $x_q$  дорівнюють нулю. Відповідно, функції  $\varphi_p$  та  $\varphi_q$  також будуть дорівнювати 0. Усі інші функції  $\varphi_i$  будуть дорівнювати 1, а, оскільки  $\varphi_i$  та  $\psi_i$  у реберних формулах поєднані диз'юнкцією, то кожна з реберних формул окрім  $f_p$  та  $f_q$  не може дорівнювати нулю.

Оскільки  $G$  є замкненим циклічним графом, то для порушення зв'язності необхідно, щоб хоча б дві з функцій, приписаних до його ребер, мають дорівнювати нулю. Зазначивши вище, що це можуть бути лише функції  $f_p$  та  $f_q$ , ми можемо показати, що виходячи з методики формування функцій  $\psi_i$ , лише одна з функцій  $\psi_p$  чи  $\psi_q$  може містити змінну  $x_q$  або  $x_p$  відповідно. Припустимо, що це функція  $\psi_p$ , тоді буде справедливим наступний вираз:

$$q \in \left[ (p+1) \bmod(n), \dots, p + \left\lfloor \frac{n-1}{2} \right\rfloor \bmod(n) \right] \quad (2.3)$$

тоді, якщо зробити підстановку усіх значень з цього проміжку до аналогічного виразу, можна побачити, що:

$$p \notin \left[ (q+1) \bmod(n), \dots, q + \left\lfloor \frac{n-1}{2} \right\rfloor \bmod(n) \right] \quad (2.4)$$



Якщо будь які з трьох довільних змінних  $x_q, x_p, x_r$  ( $q > p > r$ ) дорівнюють нулю, то кожна з функцій  $\varphi_q, \varphi_p, \varphi_r$  також буде дорівнювати нулю. Припустимо, що має місце наступне співвідношення:

$$q \in \left[ (p + 1) \bmod(n), \dots, p + \left\lfloor \frac{n-1}{2} \right\rfloor \bmod(n) \right] \quad (2.5)$$

З цього виразу витікає, що  $\psi_p$  дорівнює нулю. Рівність нулю  $\psi_q$  буде, у розглянутому випадку, еквівалентна наступному співвідношенню:

$$r \in \left[ (q + 1) \bmod(n), \dots, q + \left\lfloor \frac{n-1}{2} \right\rfloor \bmod(n) \right] \quad (2.6)$$

Однак якщо це співвідношення не виконується, можна побачити, що, оскільки  $r + \left\lfloor \frac{n-1}{2} \right\rfloor > n$ , однозначно вірним буде наступний вираз:

$$p \in \left[ (r + 1) \bmod(n), \dots, r + \left\lfloor \frac{n-1}{2} \right\rfloor \bmod(n) \right] \quad (2.7)$$

Наведене вище співвідношення еквівалентне твердженню, що  $\psi_r$  дорівнює 0. Наведену послідовність міркувань можна відтворити починаючи з кожної зі змінних, отримуючи аналогічні результати.

Для прикладу можна привести GL-модель  $K(2,7)$ , яка має наступні реберні функції:

$$\begin{aligned} f_1 &= x_1 \vee x_2 x_3 x_4 \\ f_2 &= x_2 \vee x_3 x_4 x_5 \\ f_3 &= x_3 \vee x_4 x_5 x_6 \\ f_4 &= x_4 \vee x_5 x_6 x_7 \\ f_5 &= x_5 \vee x_6 x_7 x_1 \\ f_6 &= x_6 \vee x_7 x_1 x_2 \\ f_7 &= x_7 \vee x_1 x_2 x_3 \end{aligned} \quad (2.8)$$

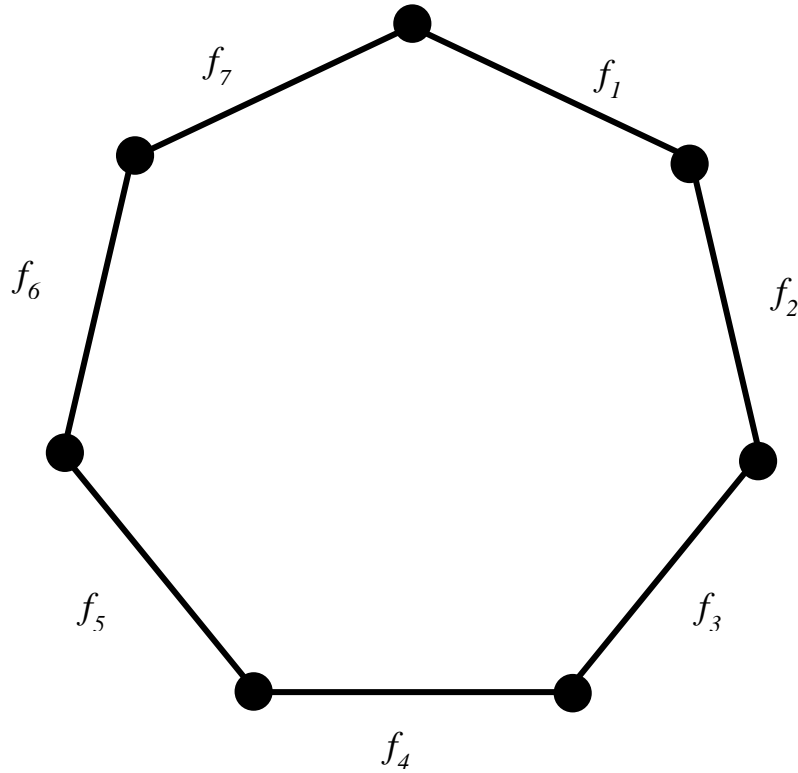


Рис. 2.1. Граф GL-моделі  $K(2,7)$

Однією з переваг такої конфігурації GL-моделей є регулярність функцій, які формуються. Не має різниці, яке з ребер буде асоційоване з певною з функцій, а будь яка функція  $f_{i+1}$  може бути отримана з функції  $f_i$  шляхом додавання одиниці до кожного з індексів змінних, що містяться у ній, та виконання операції  $\text{mod}(n)$  на кожному з цих індексів[18].

Оскільки формування такою моделі є досить простою обчислювальною задачею, використання алгоритмів такого типу було б бажаним для створення GL-моделей. Однак усі спроби узагальнити ту ж саму методику для створення  $k$ -стійких моделей виявилися марними. Поширення цього способу формування моделей, нажаль, не дає позитивних результатів.

Можна навести декілька прикладів декілька моделей з  $m > 2$ , щоб показати неоднорідність рішень. Наприклад, системи з  $m=3$  реберні функції, які виглядають наступним чином:

$$\begin{aligned}
 f_1 &= x_1 \vee x_2 \vee x_5 \vee x_6, & n &= 6 \\
 f_1 &= x_1 \vee x_2 x_3 \vee x_5 x_7, & n &= 7 \\
 f_1 &= x_1 \vee x_2 x_3 \vee x_6 x_7 x_8 x_9, & n &= 9 \\
 f_1 &= x_1 \vee x_4 x_7 x_{10} \vee x_2 x_5 x_8, & n &= 10
 \end{aligned}
 \tag{2.9}$$

$$f_1 = x_1 \vee x_2 x_3 x_4 x_5 (x_6 \vee x_7) \vee (x_8 \vee \overline{x_4} \vee \overline{x_5}) \cdot \\ \cdot (x_9 \vee \overline{x_5}) x_{10} x_{11} x_{12} x_{13} x_{13}, \quad n = 14$$

Можна побачити, що модель починає втрачати перевагу у обчислювальній складності при збільшенні кількості змінних та стає менш ефективною. Функції  $f_{i+1}$  формуються з функцій  $f_i$  так само, як і за  $m=2$ , але загальний вигляд правил формування інший.

Окрім того, у випадку  $n=14$  реберні функції GL-моделі вже не є неповторними. Використання наведеного методу для моделей з довільним  $m$  також унеможлиблюється відсутністю прямого узагальнення для будь-яких  $m$ .

### **2.1.2 Відображення підвищення та зниження надійності відмовостійких багатопроцесорних систем**

Зрозуміло, що для коректного відображення реакції обчислювальної системи в потоці відмов, недостатньо алгоритмів створення GL-моделей 2 чи 3-ВБС, оскільки реальні відмовостійкі системи зазвичай зберігають працездатність і при більшій кількості відмов елементів, з яких вони складаються. Тому існує необхідність у алгоритмах, які б мали узагальнений підхід до систем, які являють собою k-ВБС.

Однак ситуація, коли реальна система може бути представлена базовою моделлю, є ідеальною, тобто у більшості з випадків для моделювання системи необхідно створити небазову модель певного типу.

Маючи змогу використовувати вже відомі алгоритми для створення базових моделей, можна поставити завдання модифікації певних створених базових моделей для того, щоб їх поведінка в потоці відмов відповідала поведінці системи.

У загальному випадку система може бути представлена деякою непустою множиною наборів змінних, які інтерпретуються як модулі, при відмові яких система продовжує зберігати працездатність. Однак багато реальних обчислювальних систем можна представити у вигляді  $m$ -ВБС, з двома додатковими множинами  $W^+$  та  $W^-$ , де  $W^+$  складається з непустих множин від  $m+1$  до  $n-1$  змінних, що відповідають модулям, при виході яких з ладу система залишається працездатною, а  $W^-$  складається з непустих множин від 1 до  $m$

змінних, що відповідають модулям, при виході яких з ладу система втрачає працездатність.

За допомогою певних методів модифікації, можна досягти створення небазових GL-моделей з базових, таких, що вони задовольняють умови, означені вище та коректно відображають реакцію ВБС на відмови векторів з  $W^+$  та  $W$ . У поданій роботі розглядається здебільшого відображення підвищення відмовостійкості.

## 2.2 Перетворення шляхом зміни реберних функцій

Розглянемо випадок, коли  $W^+$  складається з єдиного вектору стану, за якого система зберігає роботоздатність. У найпростішому випадку цей вектор містить у собі  $m+1$  змінну, стан якої відповідає відмові відповідного модулю:

$$Y = \{x_1, x_2, \dots, x_m, x_{m+1}, x_{m+2} \dots x_{n-1}, x_n\} = \{0, 0, \dots 0, 0, 1, \dots 1, 1\} \quad (2.10)$$

Модель, яка буде коректно відображати роботу такої системи, має не тільки відповідати вимогам  $m$ -ВБС, але й відображати працездатність системи на векторі  $Y$ . Зрозуміло, що для GL-моделі це означає, що граф GL-моделі має не втрачати зв'язність при появі вектору  $Y$ .

Для створення моделей, що задовольняють ці вимоги, використовується рекурсивний метод побудови GL-моделей шляхом розподілення модулів системи на підмножини.

Його суть полягає в наступному. Розіб'ємо елементи ВБС на  $t$  підмножин  $n_1, n_2, \dots, n_t$ , кожне з яких містить  $|n_1|, |n_2|, \dots, |n_t|$  елементів, і перерахуємо всі можливі варіанти розподілу  $m$  відмов по цим підмножини. Нехай  $p$  - кількість варіантів розподілу відмов по підмножини  $n_1, n_2, \dots, n_t$ .  $m_{ij}$  - кількість відмов на  $i$ -му варіанті в  $j$ -м підмножині, що задовольняють наступне рівняння:

$$\sum_{j=0}^t m_{ij} = m, \quad i = \overline{1, p} \quad (2.11)$$

Тоді булева функція  $K_{|n_i|}^{m_{ij}}$  описує поведінку базової  $m_{ij}$ -ОМС, яка складається з  $|n_j|$  модулів з безлічі  $n_j$ , а система булевих функцій матиме такий вигляд:

$$\left\{ \begin{array}{cccc} K_{|n_1|}^{m_{11}} \vee & K_{|n_2|}^{m_{12}} \vee & \dots \vee & K_{|n_t|}^{m_{1t}} \\ K_{|n_1|}^{m_{21}} \vee & K_{|n_2|}^{m_{22}} \vee & \dots \vee & K_{|n_t|}^{m_{2t}} \\ \dots & \dots & \dots & \dots \\ K_{|n_1|}^{m_{p1}} \vee & K_{|n_2|}^{m_{p2}} \vee & \dots \vee & K_{|n_t|}^{m_{pt}} \end{array} \right. \quad (2.12)$$

Кожна з функцій  $K_{|n_i|}^{m_{ij}}$  задовольняє наступним двом нерівностям:

$$\begin{aligned} |n_j| < n, \quad j = \overline{1, t} \\ \exists i: m_{ij} < m_j, \quad j = \overline{1, t} \end{aligned} \quad (2.13)$$

Тому можна отримати спрощення первісного виразу  $K_n^m$ , шляхом подальшого розбиття отриманих  $K_{|n_i|}^{m_{ij}}$  функцій на підмножини. Критерієм зупинки рекурсивного методу є досягнення  $K_{|n_i|}^0$ ,  $K_{|n_i|}^1$  чи  $K_{|n_i|}^{|n_i|}$ .

В отриманій системі після зупинки алгоритму можна провести необхідні заміни отриманих функцій за наступними рівняннями:

$$\begin{aligned} K_{|n_i|}^0 &= 0 \\ K_{|n_i|}^1 &= \bigwedge_{k \in n_j} x_k \\ K_{|n_i|}^{|n_i|} &= \bigvee_{k \in n_j} x_k \end{aligned} \quad (2.14)$$

На прикладі таке розбиття можна проілюструвати наступним чином. Нехай існує 2-відмовостійка ВБС, що складається з 9 модулів (система  $K(2, 9)$ ). Необхідно створити GL-модель, що буде коректно відображати залежність роботоздатності цієї системи від роботоздатності її модулів.

Проведемо базове розбиття модулів системи на дві підмножини, такі, що модулі  $\{1, 2, 3, 4\}$  знаходяться у першій підмножині, а модулі  $\{5, 6, 7, 8, 9\}$  – у другій.

Таблиця 2.1

Розподілення модулів  $K(2, 9)$ -моделі на підмножини

$n_1 = \{1, 2, 3, 4\}$	$n_2 = \{5, 6, 7, 8, 9\}$
------------------------	---------------------------

Продовжимо розбиття підмножини  $n_1$  на підмножини, оскільки умова завершення алгоритму не виконана:

Таблиця 2.2

Розподілення модулів  $n_1$  на підмножини

$n_1 = \{1, 2, 3, 4\}$	
$n_{1.1} = \{1, 2\}$	$n_{1.2} = \{3, 4\}$

Після досягнення максимального розбиття, результати розбиття підмножини  $n_1$  можна звести у єдину таблицю:

Таблиця 2.3

Результати розподілення модулів  $n_1$  на підмножини

№ наборів	$n_1$	
	$n_{1.1}$	$n_{1.2}$
1	$K(1,4)$	
2	$K(2,2)$	-
3	$K(1,2)$	$K(1,2)$
4	-	$K(2,2)$
5	-	-
6	-	-
7	-	-
8	-	-

Тепер зосередимось на розбитті підмножини  $n_2$ :

Таблиця 2.4

Розподілення модулів  $n_2$  на підмножини

$n_2 = \{5, 6, 7, 8, 9\}$		
$n_{2.1} = \{5, 6\}$	$n_{2.2} = \{7, 8\}$	$n_{2.3} = \{9\}$

Після досягнення максимального розбиття, результати розбиття підмножини  $n_1$  можна звести у єдину таблицю:

Таблиця 2.5

Результати розподілення модулів  $n_1$  на підмножини

№ наборів	$n_2$		
	$n_{2.1}$	$n_{2.2}$	$n_{2.3}$
1	$K(1,5)$		
2	-	-	-
3	-	-	-
4	-	-	-
5	$K(2,2)$	-	-
6	$K(1,2)$	$K(1,3)$	
7	-	$K(1,2)$	$K(1,1)$
8	-	$K(2,2)$	-

На основі отриманих розподілень можна отримати наступну таблицю спільних результатів, яку можна використовувати для побудови функцій GL-моделі:

Таблиця 2.6

Фінальні результати розподілення модулів моделі  $K(2, 9)$  на підмножини

№ наборів	$n_1$		$n_2$		
	$n_{1.1}$	$n_{1.2}$	$n_{2.1}$	$n_{2.2}$	$n_{2.3}$
1	$K(1,4)$		$K(1,5)$		
2	$K(2,2)$	-	-	-	-
3	$K(1,2)$	$K(1,2)$	-	-	-

4	-	$K(2,2)$	-	-	-
5	-	-	$K(2,2)$	-	-
6	-	-	$K(1,2)$	$K(1,3)$	
7	-	-	-	$K(1,2)$	$K(1,1)$
8	-	-	-	$K(2,2)$	-

На підставі даної таблиці запишемо булеві функції:

$$f_1 = x_1 x_2 x_3 x_4 \vee x_5 x_6 x_7 x_8 x_9$$

$$f_2 = x_1 \vee x_2$$

$$f_3 = x_1 x_2 \vee x_3 x_4$$

$$f_4 = x_3 \vee x_4$$

$$f_5 = x_5 \vee x_6$$

$$f_6 = x_5 x_6 \vee x_7 x_8 x_9$$

$$f_7 = x_7 \vee x_8 x_9$$

$$f_8 = x_8 \vee x_9$$

(2.15)

Відповідно, GL-модель буде мати наступний вигляд:

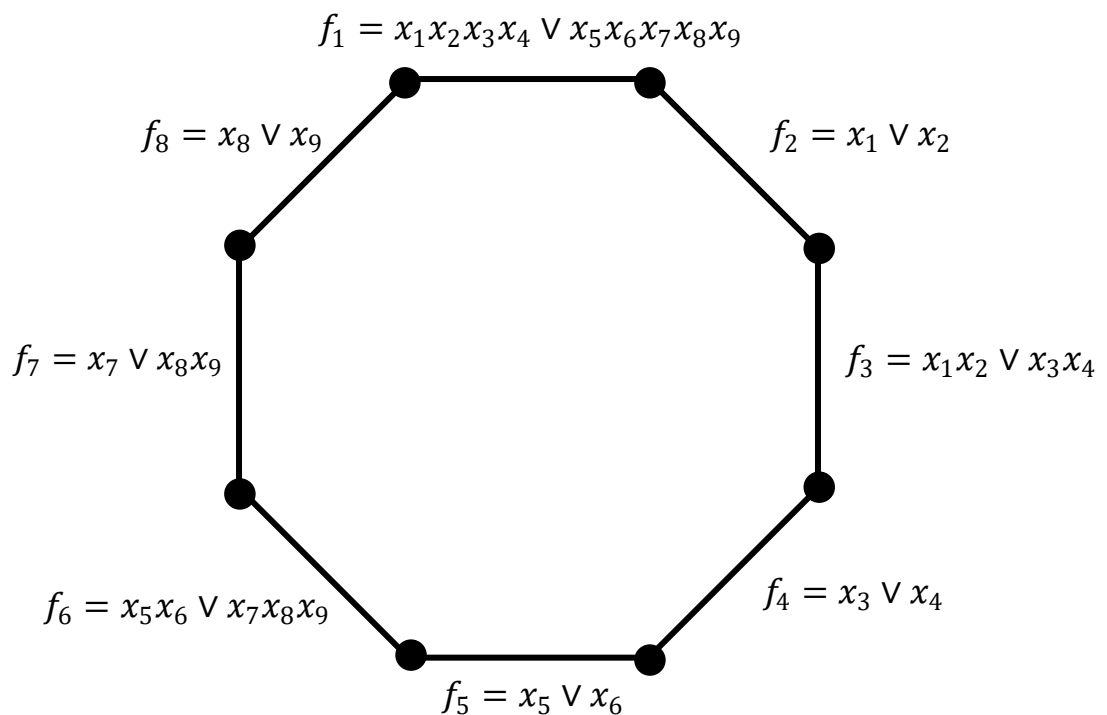


Рис. 2.2 GL-модель  $K(2,9)$  системи



Для того, щоб граф циклічної GL-моделі не втрачав зв'язності при появі вектора стану  $Y$ , необхідно, щоб максимум одна функція приймала нульовий значення на цьому векторі стану. Якщо  $p$  – кількість функцій, рівних 0 при появі вектора стану  $Y$ , але необхідно перетворити будь-які  $p-1$  функцій наступним чином:

$$f'_i = f_i \vee x = \bar{x}_1 \bar{x}_2 \cdot \dots \cdot \bar{x}_{m+1} x_{m+2} \cdot \dots x_n \quad (2.16)$$

При покроковій зміні моделі одне і те ж ребро може пропадати при появі багатьох векторів стану із заданої множини. Тому після зміни моделі все модифіковані функції має сенс мінімізувати за допомогою будь-якого відомого методу мінімізації булевих функцій.

Для модифікації базових моделей 2-відмовостійких ВБС існують більш ефективні методи, що базуються на методі створення GL-моделей, розглянутому у розділі 2.1.

Для випадку, коли необхідно модифікувати 2-ВБС, додавши певні множини змінних потужністю 3, що відповідають модулям, при виході з ладу яких система зберігає працездатність, можна використати алгоритм для трансформації моделей, що відображають реакцію ВБС, стійких до двох відмов, на появу заданої кількості трьохмодульних відмов.

Для цього кожен вектор стану, що містить три відмови та має належати до множини додаткової стійкості  $W^+$  представляють у вигляді  $r'$ -векторів, або векторів відстаней. Елементами таких векторів є кількості одиниць між нулями у векторі стану (наприклад, вектору стану ВБС, що складається з 9 модулів, значення змінних котрого дорівнюють (001110111), відповідає  $r'$ -вектор (0, 3, 3).

Для випадку системи, що стійка до відмови двох модулів,  $W^+$  буде складатись з двох підмножин:

- $W^+_2$ , група векторів, при появі яких з графу базової моделі видаляється 2 ребра.
- $W^+_3$ , група векторів, при появі яких з графу базової моделі видаляється 3 ребра.

Зрозуміло, що для випадку, який розглядається, вектори  $r'$  мають вигляд  $(r_1, r_2, r_3)$ . Множина  $W^+_3$  буде складатись з векторів стану, де  $r_i < \sigma - 1$  для  $i=1,2,3$ ,

де  $\sigma = \left\lceil \frac{n-1}{2} \right\rceil$ ,  $W_2^+$  буде вміщати усі вектори, що залишилися. Зрозуміло, що у векторах з  $W_2^+$  хоча б один  $r_i > \sigma - 1$  хоча б для одного з  $i=1,2,3$ .

Сам спосіб модифікації моделі пропонує модифікувати лише певні функції ребер графу, не змінюючи при цьому його структури. Це має сенс, оскільки допомагає уникнути додаткової обчислювальної складності у кінцевій моделі.

Покажемо, що перетворивши реберну функцію  $h_i$  у функцію  $h'_i = h_i \vee \overline{x_{i-k}}$ , де  $k \in [1, \sigma]$ , то вона перестане приймати нульове значення на  $\sigma$  векторах стану. Припустимо, що функція

$$h_i = x_i \vee x_{i+1} \dots x_{i+\sigma} \quad (2.17)$$

була змінена наступним чином:

$$h'_i = x_i \vee x_{i+1} \dots x_{i+\sigma} \vee \overline{x_{i-k}}, k \in [1, \sigma] \quad (2.18)$$

Позначимо поєднану кон'юнкцією частину  $h_i$  як  $T_i$ , тоді при  $x_i=0$ ,  $x_{i-k}=0$  и  $x_j=0$  ( $k \in T_i$ ) функція  $h_i=0$ , а значення функції  $h'_i=1$ . Тоді зрозуміло, що кількість векторів стану, за яких ребро, позначене функцією  $h'_i$ , які не будуть видалені з графу, дорівнює кількості змінних  $T_i$ , тобто  $\sigma$ .

Назвемо множину векторів стану з трьома відмовами, при появі яких функція  $h'_i$  перестане приймати нульове значення  $W_i^*$ . Вектори з цієї множини можуть належати як до  $W_2^+$ , так і до  $W_3^+$ , однак ті з них, що входять до  $W_2^+$ , будуть блокуватися. Позначимо ці вектори  $W_i^b$ , а усі інші, що раніш входили до групи  $W_3^+$  тепер увійдуть до групи  $W_2^+$ ; їх позначимо  $W_i^0$ .

Розглянемо вектори стану, що належать до  $W_i^*$ . Складова вектору  $r_1$  відстаней відповідає кількості одиниць між  $x_i$  та  $x_j$ , де  $x_j \in T_i, j \in [i+1, i+\sigma]$ . Тому можна однозначно сказати, що  $r_1 \in [1, \sigma - 1]$  для кожного з цих векторів з усієї множини  $W_i^*$ .

Таким самим чином елемент  $r_3$ , що відповідає кількості змінних між  $x_{i-k}$  та  $x_i$ ,  $k \in [1, \sigma]$ , що визначає  $r_3 \in [0, \sigma - 1]$ , та  $r_3 = k - 1$ .

Можна зазначити, що  $r_3 \in [n - m - \sigma - 1, n - m - 2]$ . Зазначимо, що множина векторів стану  $W_i^*$  виражено наступною послідовністю векторів:

$$(0, n - m - 2, m - 1), (1, n - m - 3, m - 1), \dots, (\sigma - 1, n - m - \sigma - 1, m - 1)$$

Вище було зазначено, що коли вектор відстаней відповідає умові  $r_i < \sigma - 1$ , то він належить до  $W^+_3$ . Можна побачити, що складові вектору  $r'$   $r_1$  та  $r_3$  завжди відповідають цій умові, тому саме  $r_2$  визначає, чи належать вектори стану, що відповідають поданому  $r'$ , до  $W_i^0$ .

Складова вектору  $r_2 = n - k - \sigma - 1$  задовольняє умову  $r_i < \sigma - 1$  при  $m \geq 2$  для моделей з парним  $n$ , та  $m \geq 1$  для моделей з непарним  $n$ . Якщо провести аналогічний аналіз для усіх векторів з наведеної вище послідовності, можна побачити, що при використанні такого методу модифікації функції для моделей з парною кількістю модулів останні  $m - 1$  функцій з наведеної вище послідовності належать до множини  $W_i^0$ , а для моделей з непарною кількістю модулів останні  $k$  функцій з наведеної вище послідовності належать до множини  $W_i^0$ .

Отже, модифікація однієї функції  $h_i$  за подібним методом призводить до того, що:

- для моделей з непарним  $n$ ,  $|W_i^b| = \sigma - k$ ,  $|W_i^0| = m$
- для моделей з парним  $n$ ,  $|W_i^b| = \sigma - k + 1$ ,  $|W_i^0| = m - 1$

Перевагою використання методів модифікації GL-моделей шляхом зміни їх функцій є той факт, що такі перетворення не ускладнюють структури графа GL-моделі, тому за використання цих методів вирішення задачі визначення зв'язності графа є тривіальним. Однак, для підвищення ефективності такі методи потребують використання алгоритмів спрощення

### 2.3 Перетворення шляхом введення додаткових ребер

Оскільки GL-моделі поєднують у собі використання властивостей графів та булевих функцій, то змінити реакцію моделі на вектори стану системи можна не лише за допомогою модифікації реберних функцій, а й за допомогою зміни структури графу, з якого модель складається. Зрозуміло, що для відображення підвищення відмовостійкості системи, граф моделі повинен зберігати зв'язність на певних векторах стану, за яких граф базової моделі втрачає зв'язність. Тому, метод перетворення базових GL-моделей у небазові з метою відображення підвищення відмовостійкості використовує додавання внутрішніх ребер до графу моделі, які блокують втрату зв'язності у тих випадках, коли певні ребра базового графу

зникають з нього через те, що їх реберні функції дорівнюють нулю на певних векторах стану з множини  $W^+$ .

Розглянемо приклад додавання внутрішнього ребра для відображення GL-моделлю підвищення відмовостійкості системи. Візьмемо за базову модель  $K(2, 8)$  GL-модель:

Таблиця 2.7

Формування базової моделі  $K(2, 8)$

№ наборів	$\{1,2,3,4\}$		$\{5,6,7,8\}$	
1	$K(1,4)$		$K(1,4)$	
2	$K(2,2)$	-	-	-
3	$K(1,2)$	$K(1,2)$	-	-
4	-	$K(2,2)$	-	-
5	-	-	$K(2,2)$	-
6	-	-	$K(1,2)$	$K(1,2)$
7	-	-	-	$K(2,2)$

На основі таблиці побудуємо базову GL-модель на основі кільцевого графу з 7 ребрами:

$$\begin{aligned}
 f_1 &= x_1 x_2 x_3 x_4 \vee x_5 x_6 x_7 x_8 \\
 f_2 &= x_1 \vee x_2 \\
 f_3 &= x_1 x_2 \vee x_3 x_4 \\
 f_4 &= x_3 \vee x_4 \\
 f_5 &= x_5 \vee x_6 \\
 f_6 &= x_5 x_6 \vee x_7 x_8 \\
 f_7 &= x_7 \vee x_8
 \end{aligned}
 \tag{2.19}$$



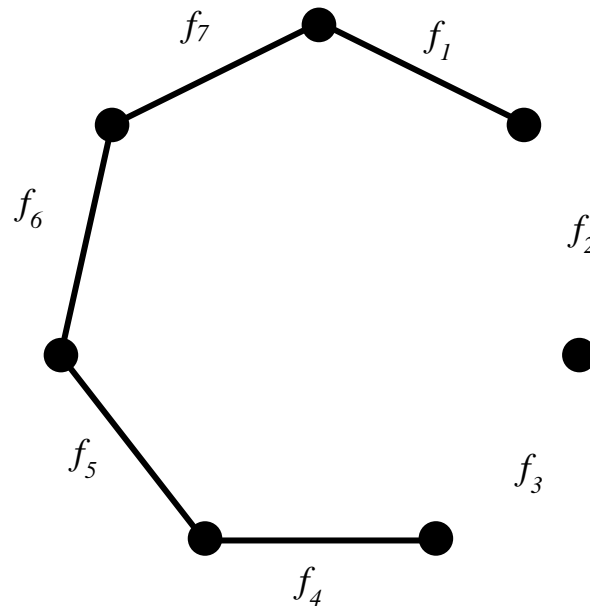


Рис. 2.5 Стан  $K(2,8)$  GL-моделі для вектору  $w_2$

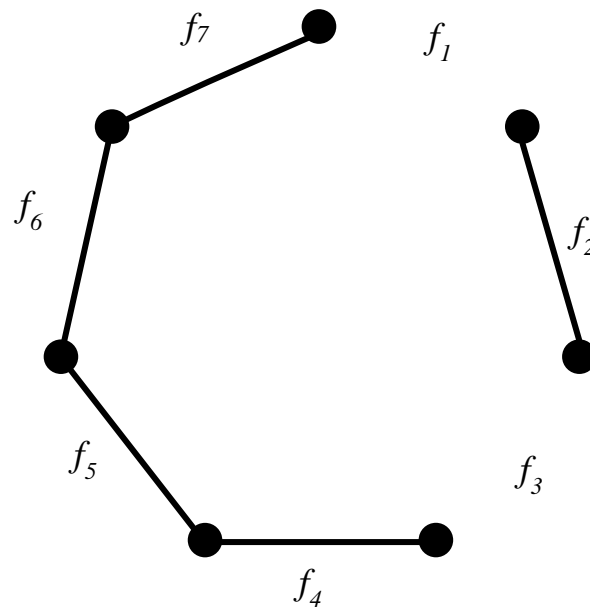


Рис. 2.6 Стан  $K(2,8)$  GL-моделі для вектору  $w_3$

Можна побачити, що для векторів стану для векторів  $w_1$  та  $w_2$  ребра, що зникають з графу, належать одним і тим же функціям. Зв'язність у цих двох станах втрачається шляхом втрати з'єднуючих ребер до єдиної вершини. Зв'язність у стані для вектору  $w_3$  втрачається шляхом відокремлення ребра  $f_2$  від графу.

Оскільки вершина, що відокремлюється у станах для векторів  $w_1$  та  $w_2$  належить ребру  $f_2$ , то одне додаткове ребро до цієї вершини від будь-якої іншої

вершини з частини графу, що не відокремлюється при надходженні вектору стану  $w_3$  буде достатньо для того, щоб блокувати втрату зв'язності графом на цих трьох векторах стану.

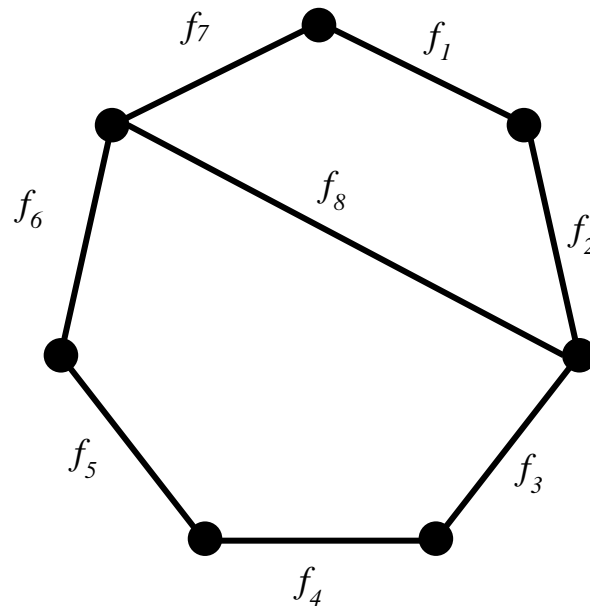


Рис. 2.7 Модифікована  $K(2,8)$  GL-модель

Тоді при надходженні на вхід моделі векторів стану  $w_1$  та  $w_2$  структура графу буде мати наступний вигляд:

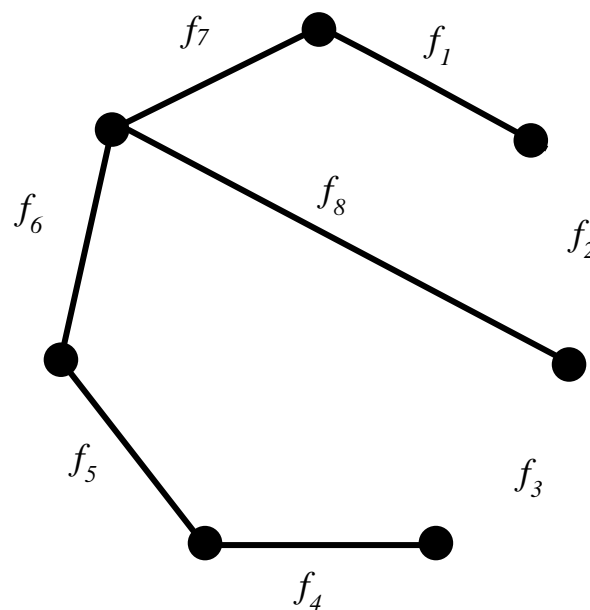


Рис. 2.8. Стан модифікованої  $K(2,8)$  GL-моделі для векторів  $w_1$  та  $w_2$

При надходженні на вхід моделі вектору стану  $w_2$  структура графу набуває наступного вигляду:

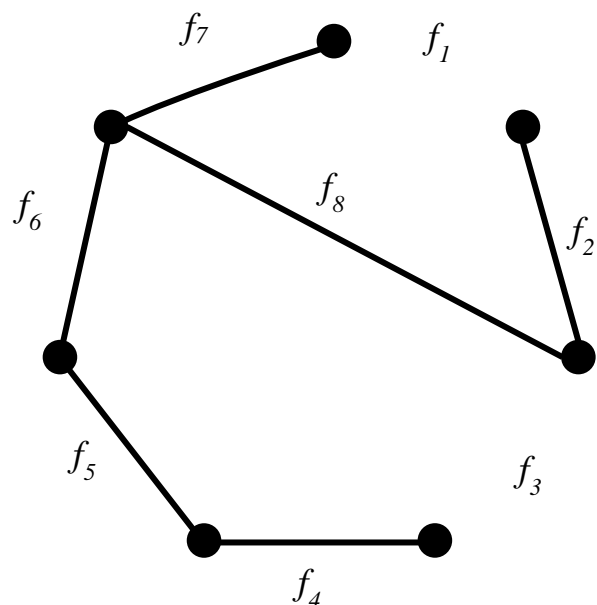


Рис. 2.9. Стан модифікованої  $K(2,8)$  GL-моделі для векторів  $w_3$

Таким чином, додаткове ребро допомагає зберегти зв'язність у графі при надходженні відповідних векторів стану. Однак, для того, щоб забезпечити ідентичність відображення стану системи модифікованою моделлю базовій, необхідно підібрати відповідне значення функції  $f_8$ .

У загальному випадку, такі додаткові ребра мають бути присутні у графі лише при надходженні відповідних векторів стану, відмову модулів з яких ці додаткові ребра блокують. Для вектору стану, що блокується одним додатковим ребром, функцію цього ребра можна прирівняти конституенті одиниці цього вектору. Наприклад, для блокування відмови модулів, що трапляються у векторі стану  $w_1$ , реберна функція додаткового внутрішнього ребра мала би наступний вигляд:

$$f = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 x_5 x_6 x_7 x_8 \quad (2.21)$$

Оскільки у поданому прикладі блокування трьох векторів стану відбувається одним й тим самим ребром, то значення функції має дорівнювати диз'юнкції конституент одиниці кожного з векторів стану  $w_1$ ,  $w_2$  та  $w_3$ :

$$f = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 x_5 x_6 x_7 x_8 \vee \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 x_5 x_6 x_7 x_8 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \bar{x}_5 x_6 x_7 x_8 \quad (2.22)$$

У загальному випадку, визначення мінімальної кількості внутрішніх ребер, що необхідно провести у графі GL-моделі, за відомими векторами стану у множині  $W^+$  є нетривіальною задачею. Для рішення цієї проблеми



використовується метод розподілення ребер графу на так звані  $S$ -підмножини.  $S$ -підмножини – 2 чи більше непустих непересічних підмножин основних ребер графа GL-моделі, якщо зв'язність графа зберігається при втраті будь-якої пари ребер, що належать різним підмножини, і якщо зв'язність графа порушується при втраті будь-якої пари ребер, що належать будь-якого з підмножин, що містить більше одного ребра.

Розглянемо приклад : додаткове внутрішнє ребро проведене у графі GL-моделі, множина ребер графу позначена як  $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ .

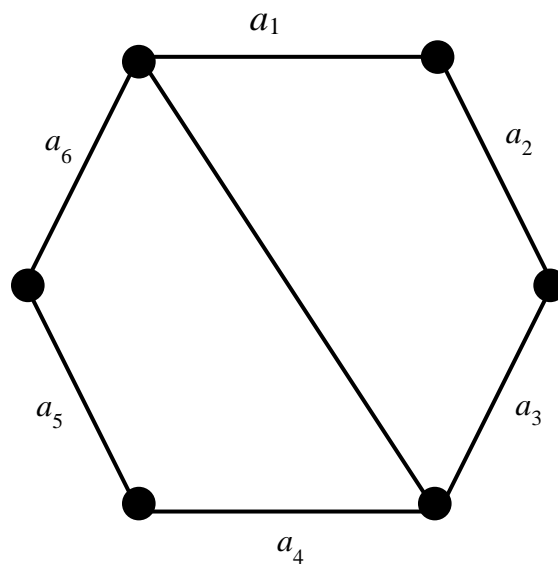


Рис. 2.10. Граф GL-моделі  $K(2,6)$  з додатковим ребром

Додаткове ребро ділить множину  $A$  на дві  $S$ -підмножини,  $A_1$  та  $A_2$ . До підмножини  $A_1$  входять ребра  $\{a_1, a_2, a_3\}$ , до підмножини  $A_2$  входять ребра  $\{a_4, a_5, a_6\}$ .

Будь-яке додаткове внутрішнє ребро буде поділяти вже існуючі  $S$ -підмножини на додаткові підмножини, збільшуючи кількість можливих комбінацій ребер, видалення пар яких не призводять до втрати зв'язності у графі. Розглянемо додавання додаткового ребра до вже модифікованої системи, наведеної вище:

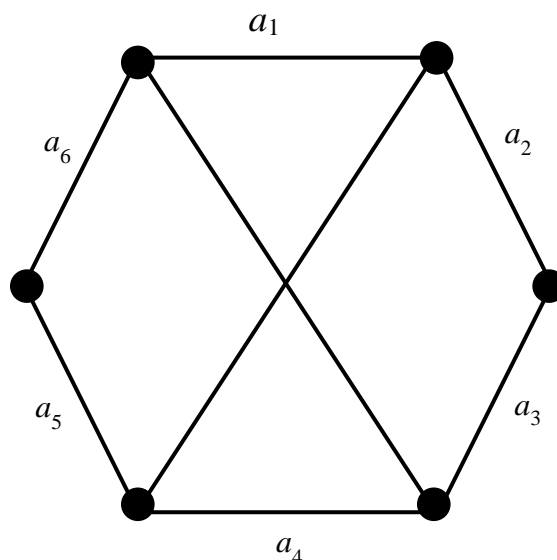


Рис. 2.11. Модифікований граф GL-моделі  $K(2,6)$  з додатковим ребром

Кількість  $S$ -підмножин після додавання другого ребра подвоїлась, розділивши ребра на 4  $S$ -підмножини:  $=\{a_1\}, \{a_2, a_3\}, \{a_4\}, \{a_5, a_6\}$ .

Можливе розбиття графу GL-моделі на довільну кількість  $S$ -підмножин, що не перевищує кількість основних ребер. За  $S$ -підмножини легко отримати множини «захищених» пар ребер (під «захищеною» парою мається на увазі пара основних ребер, при зникненні яких зв'язність графа зберігається).

Зрозуміло, що для забезпечення зв'язності на заданих наборах при втраті максимум двох ребер для кожного вектору стауну можна побудувати граф  $A^*$ , упорядкувавши ребра графа  $A$  і провівши додаткові ребра, тим самим розділивши основні на  $S$ -підмножини так, щоб вершини кожної з пар ребер, що зникають за відповідного вектору стану, перебували в різних підмножинах. Функції, приписані додатковим ребрам, повинні формуватися таким чином, щоб зв'язність графа, при появі інших наборів, залишалася відповідної базової моделі[18].

Однак, для визначення мінімальної необхідної кількості додаткових ребер (та  $S$ -підмножин) існують більш ефективні рішення.

Для пошуку оптимального розділення ребер графу GL-моделі на  $S$ -підмножини, необхідно побудувати допоміжний граф  $V$ , множина вершин в якому складається з ребер базового графу, а ребра з'єднують між собою ті пари вершин, які зникають з графу при надходженні до моделі відповідних векторів стану з множини  $W^+$ . Розглянемо побудову допоміжного графу для базової

моделі  $K(2, 8)$ , наведеної вище. Нехай множина  $W^+$  містить наступні вектори стану:

$$\begin{aligned} w_1 &= \{0,0,0,1,1,1,1,1\} \\ w_2 &= \{0,1,0,1,0,1,1,1\} \\ w_3 &= \{1,1,1,0,0,1,0,1\} \end{aligned} \quad (2.23)$$

Розглянемо конфігурації графу, що виникають за кожного з зазначених вище векторів стану:

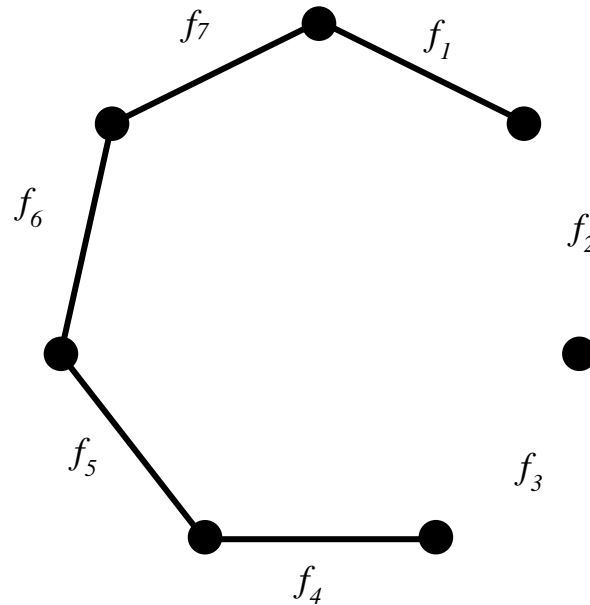


Рис. 2.12. Стан  $K(2,8)$  GL-моделі для вектору  $w_1$

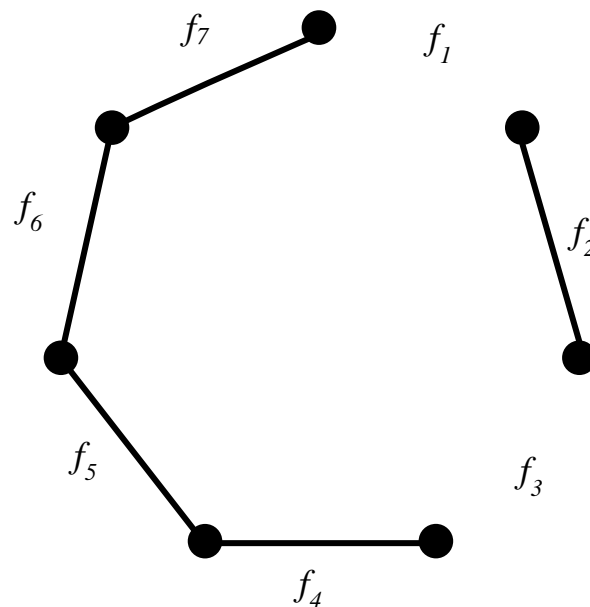


Рис. 2.13. Стан  $K(2,8)$  GL-моделі для вектору  $w_2$

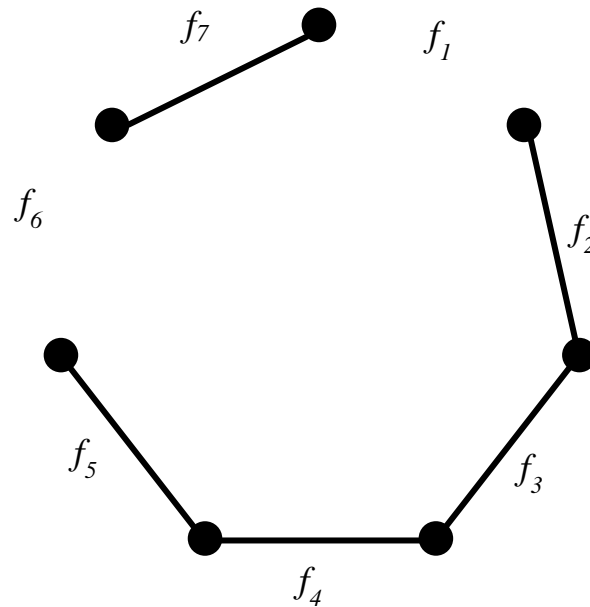


Рис. 2.14. Стан  $K(2,8)$  GL-моделі для вектору  $w_3$

Відповідні пари функцій, що дорівнюють нулю за надходження кожного з цих векторів, відповідно:  $\{f_2, f_3\}$ ,  $\{f_1, f_3\}$ ,  $\{f_1, f_6\}$ . Побудуємо  $V$ -граф базуючись на цих даних:

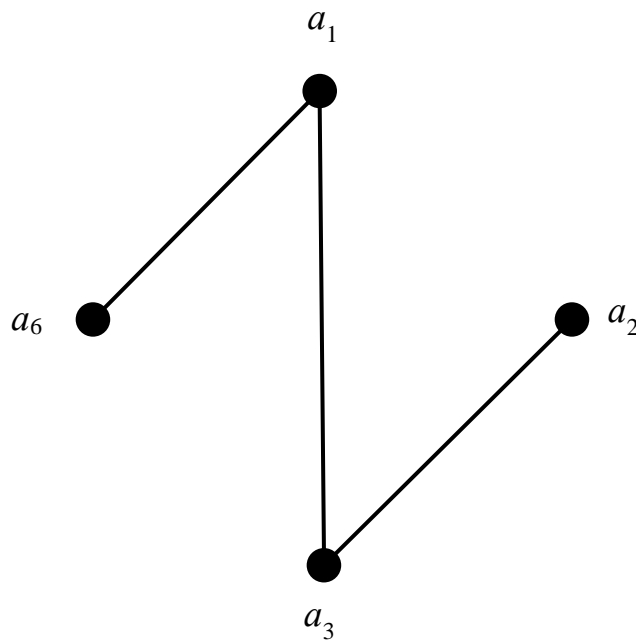


Рис. 2.15.  $V$ -граф для модифікації моделі  $K(2,8)$

Визначення  $S$ -підмножин ребер графа  $A$  еквівалентно визначенню внутрішньо стійких множин вершин графа  $V$  (внутрішньо стійким безліччю вершин називається підмножина вершин графа не пов'язаних між собою). Таким чином, знаходження розбиття множини вершин графа  $V$  на мінімальне число

внутрішньо стійких множин еквівалентне визначенню шуканих  $S$ -підмножини ребер графа  $A$ .

Таким чином задача пошуку  $S$ -підмножин графа  $A$  може бути зведена до розфарбуванні вершин графа  $V$ . Розфарбування вершин графу є окремим випадком розмітки графів. При розфарбовуванні елементам графа ставляться у відповідність мітки з урахуванням певних обмежень; ці мітки традиційно називаються «квітами». У найпростішому випадку такий спосіб забарвлення вершин графа, при якому будь-яким двом суміжним вершинам відповідають різні кольори. Вершини, які отримали один колір, будуть належати одній внутрішньо стійкій множини.

Для отримання швидкої приблизної оцінки кількості можна використати відомий алгоритм жадібного розфарбування графа.

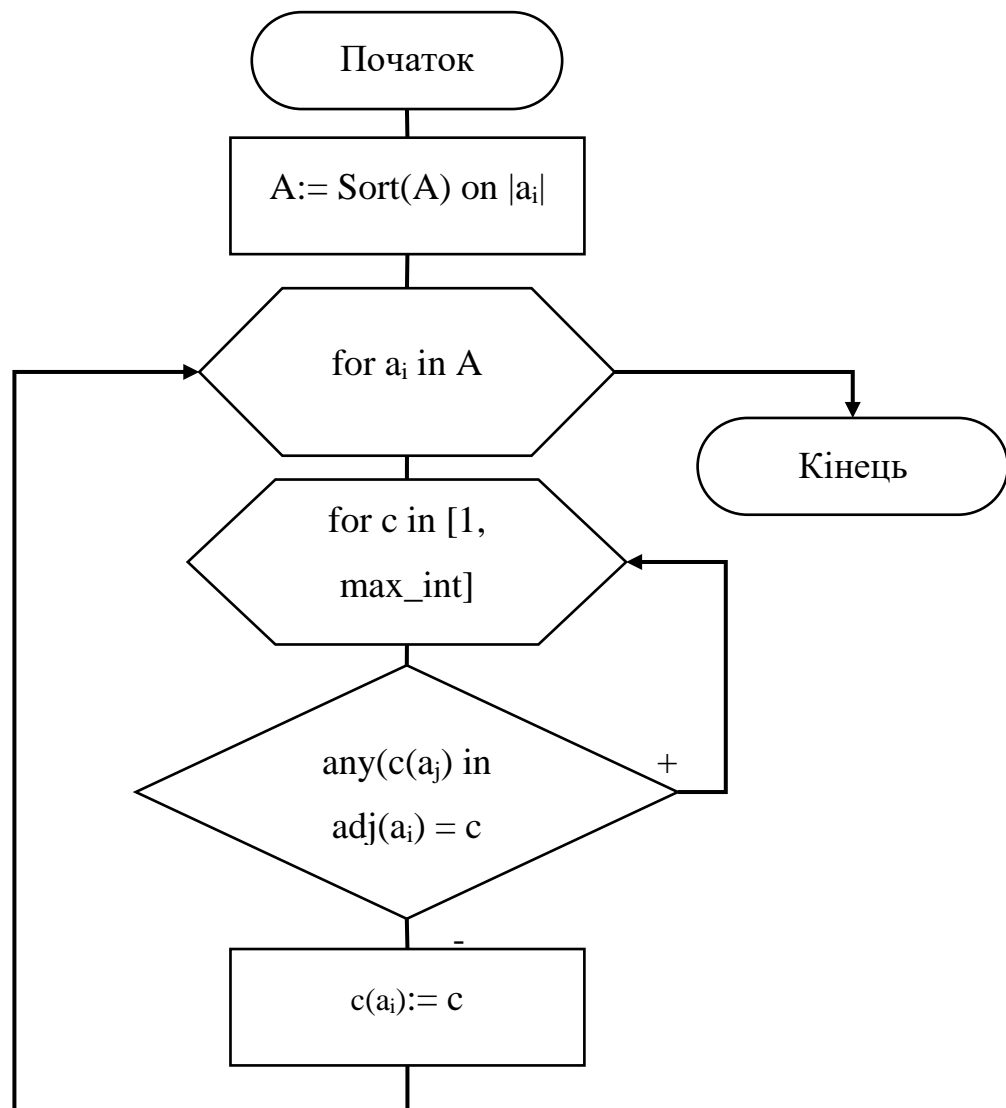


Рис. 2.16. Блок-схема алгоритму жадібного розфарбування графу

Позначимо кількість кольорів, або, що те ж саме, кількість отриманих непересічних внутрішньо стійких множин як  $\lambda$ . Отримані множини внутрішньо стійких множин позначається  $s_i = \{a_{i1}, a_{i2}, \dots, a_{ij}\}$ ,  $S = \{s_1, s_2, s_3 \dots s_\lambda\}$ .

Мінімальна кількість таких підмножин дорівнює хроматичному числу графа  $V$ . Кількість варіантів розбиття на підмножини визначається значенням хроматичного полінома графа  $V$  за значенням  $\lambda$ .

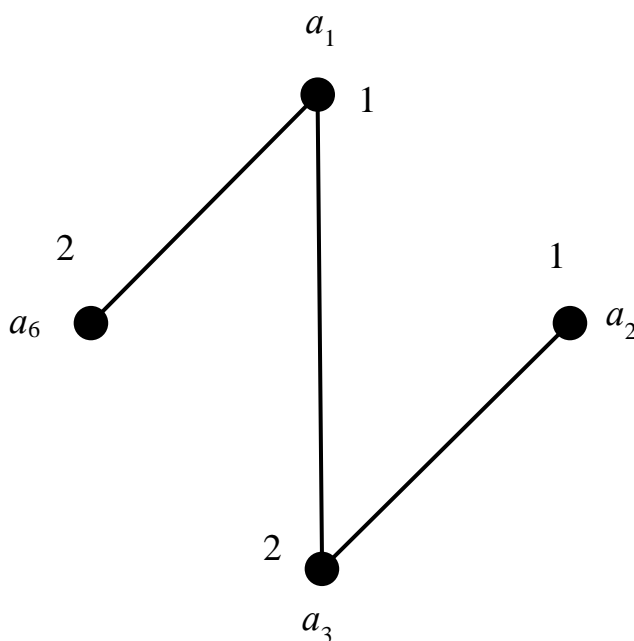


Рис. 2.17. Розфарбований  $V$ -граф для модифікації моделі  $K(2,8)$

Кожне додаткове ребро може бути проведене у графі  $A$  таким чином, що кількість  $S$ -підмножин збільшується у два рази, окрім останнього ребра за випадку, коли  $\lambda$  – непарне. Необхідна і достатня кількість додаткових ребер для найгіршого випадку:

$$e = \left\lfloor \frac{\lambda + 1}{2} \right\rfloor \quad (2.24)$$

Вузли одного кольору належать до однієї  $S$ -підмножини, яка необхідна для блокування відмов у графі  $GL$ -моделі. Для пошуку оптимального місця для додавання внутрішніх ребер необхідно перемістити ребра графу (користуючись тим, що базова  $GL$ -модель не накладає обмежень на взаємне положення ребер у графі) таким чином, що усі ребра, що належать одній й тій самій стійкій підмножині, утворювали ланку.

Розглянемо метод обирання внутрішніх ребер для додавання на наведеному вище прикладі:

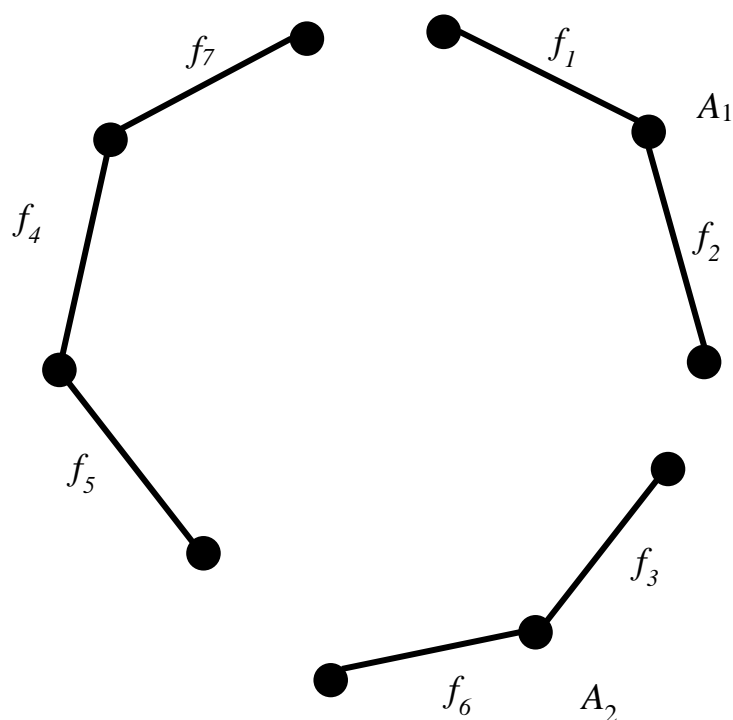


Рис. 2.18. Граф GL-моделі розділений на S-підмножини

Розподіливши граф таким чином, замінимо ребра з S-підмножин на абстрактне представлення ланок S-підмножин:

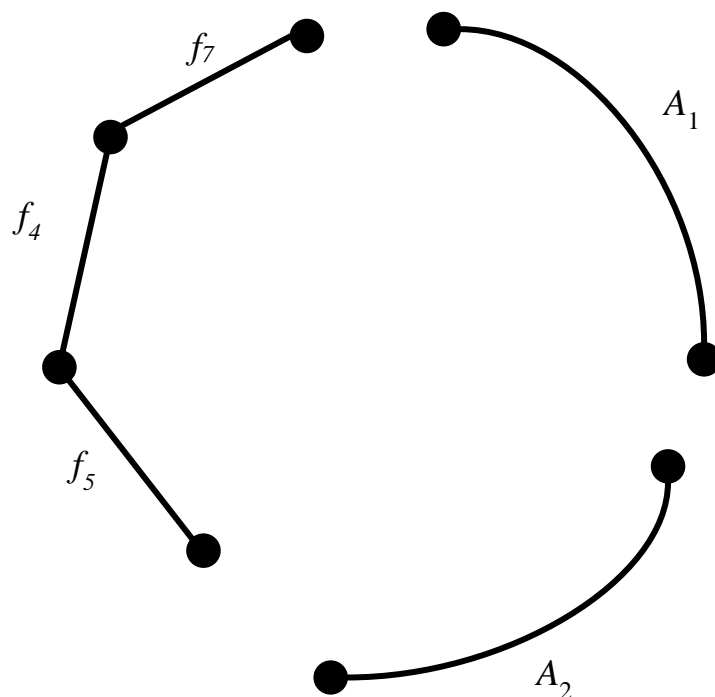


Рис. 2.19. Граф GL-моделі розділений на S-підмножини

Після цього проведення додаткових внутрішніх ребер є тривіальною задачею: необхідно провести ребро від будь-якої вершини, що не включена в жодну з  $S$ -підмножин, до вершини, яка з'єднує ланки  $S$ -підмножин; або, якщо усі вершини (іх відповідні ребра) належать до певної  $S$ -підмножини, то провести додаткові ребра з'єднуючи вузли, що з'єднують ланки  $S$ -підмножин.

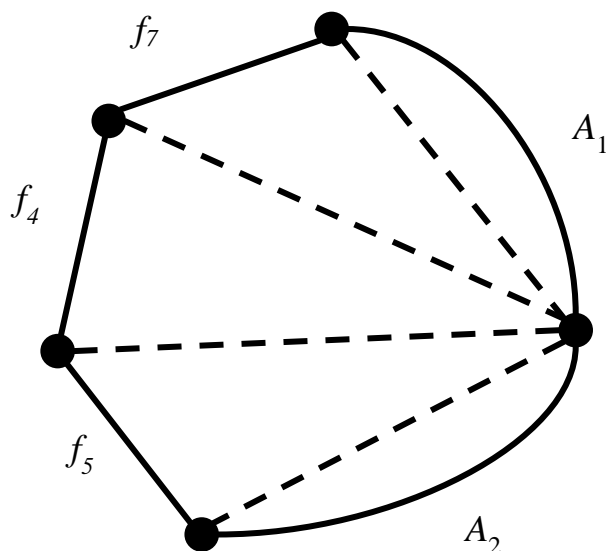


Рис. 2.20. Можливі додаткові ребра на графі GL-моделі, розділеної на  $S$ -підмножини

Заміна ланцюгу одним ребром не обмежує спільності, оскільки в одній  $S$ -підмножині не може одночасно зникнути більше одного ребра при будь-якому наборі з  $W^+$  (з побудови  $S$ -підмножин).

Оскільки при переході до графу  $S$  ми отримали граф, в якому кількість  $S$ -підмножин і ребер збігається, загальний випадок зводиться до розглянутого вище. Додаткові ребра можна проводити, модифікуючи безпосередньо граф  $S$ , так як можливий зворотний перехід, шляхом заміни ребер графа  $S$  вихідними ланцюгами.

Кінцевим результатом застосування методу до наведеного вище прикладу буде наступна небазова GL-модель:



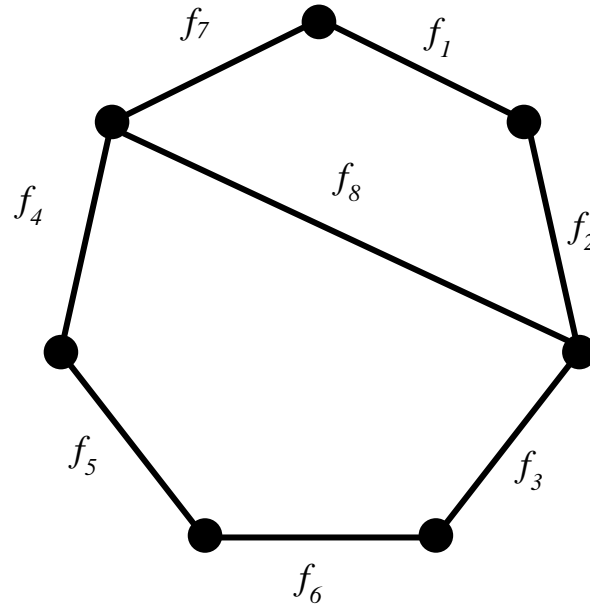


Рис. 2.21. Небазова GL-модель, отримана шляхом перетворення за методом додавання внутрішніх ребер

$$\begin{aligned}
 f_1 &= x_1 x_2 x_3 x_4 \vee x_5 x_6 x_7 x_8 \\
 f_2 &= x_1 \vee x_2 \\
 f_3 &= x_1 x_2 \vee x_3 x_4 \\
 f_4 &= x_3 \vee x_4 \\
 f_5 &= x_5 \vee x_6 \\
 f_6 &= x_5 x_6 \vee x_7 x_8 \\
 f_7 &= x_7 \vee x_8
 \end{aligned}
 \tag{2.25}$$

$$f_8 = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 x_5 x_6 x_7 x_8 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \bar{x}_5 x_6 x_7 x_8 \vee x_1 x_2 x_3 \bar{x}_4 \bar{x}_5 x_6 \bar{x}_7 x_8$$

Загальна схема алгоритму перетворення базових GL-моделей в небазові, заснованого на наведеному вище методі, можна побачити на рисунку 2.22.

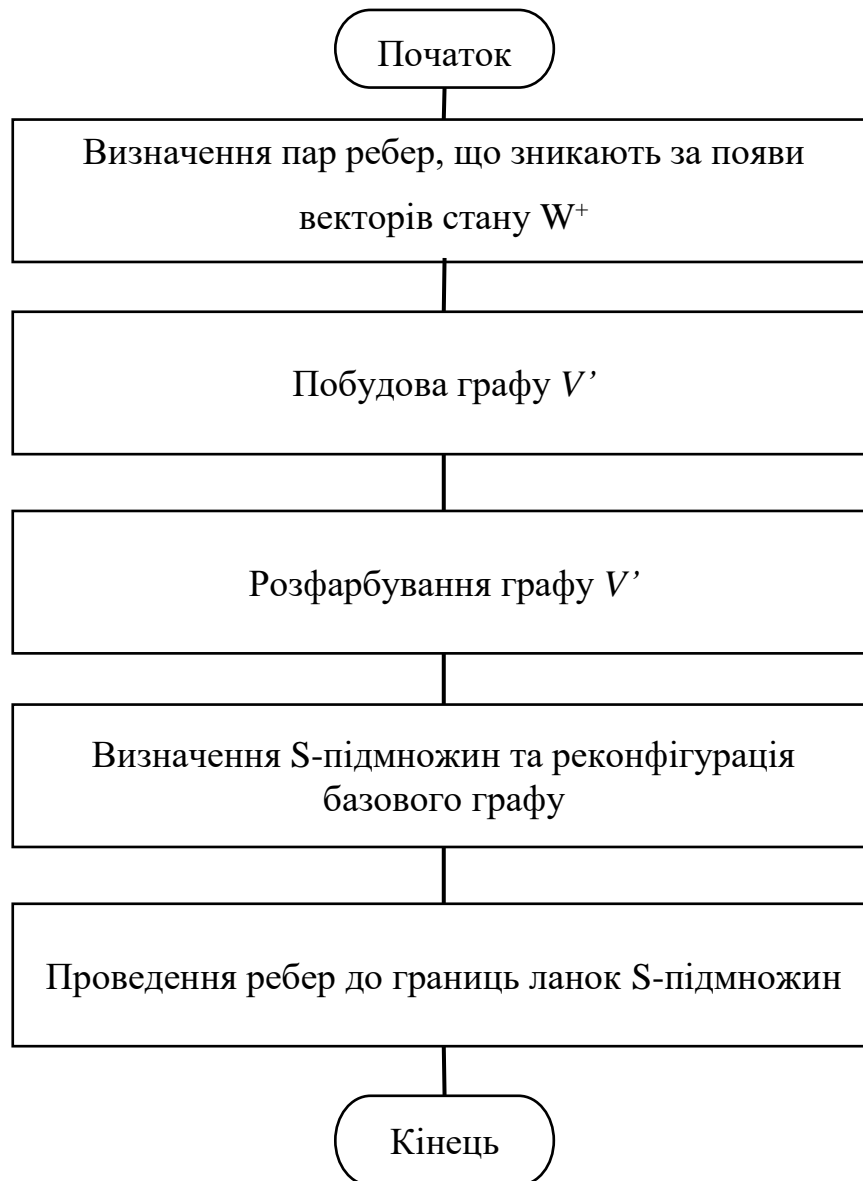


Рис. 2.22. Блок-схема алгоритму модифікації GL-моделі шляхом додавання внутрішніх ребер

Головною проблемою використання такого алгоритму є сама проблема визначення  $S$ -підмножин, оскільки алгоритми, що точно визначають розфарбування графу, є NP-повними і тому пошук  $S$ -підмножин у графі може займати значний час. Наведений алгоритм жадібного розфарбування графу може бути застосований для розрахунку приблизних значень, призводить до некоректних результатів на певних графах, наприклад графі конфігурації «корона», що у найкращому випадку можуть бути розфарбовані двома кольорами, жадібний алгоритм розфарбовує  $n$  кольорами.

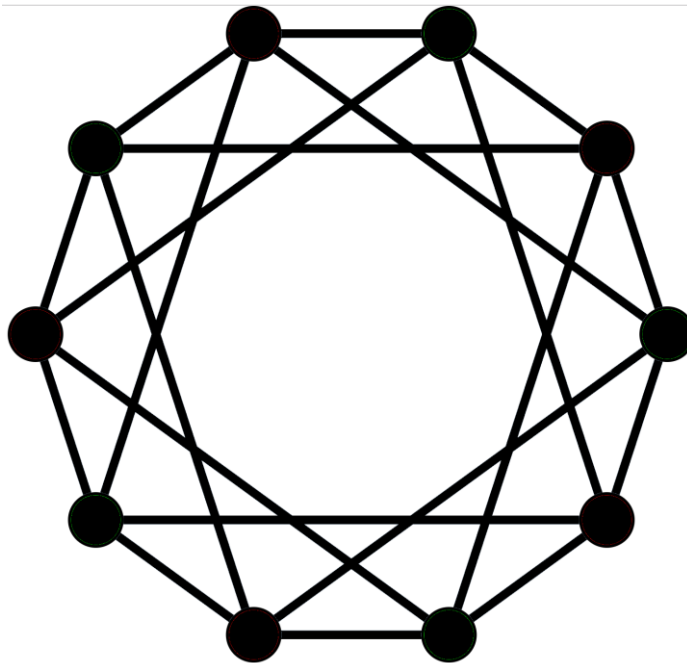


Рис. 2.23. Корона з 10 вершинами

Окрім того, визначення зв'язності графу для обчислення стану системи на основі стану її модулів, стає складнішим з додаванням внутрішніх ребер, тому попередній аналіз структури внутрішніх ребер, що буде необхідна для модифікації, та їх кількості, є важливою для успішного застосування цього методу.

## **Висновки розділу 2**

У розділі розглядається проблема створення небазових GL-моделей, аналізуються методи перетворення базових моделей у небазові. Аналіз двох головних підходів до модифікації дає оцінку їх можливого використанню для створення моделей реальних відмовостійких систем.

Під час аналізу були наведені переваги та недоліки вищезазначених методів. Особливо відзначена важливість аналізу структур, що виникають під час застосування методу перетворення базових GL-моделей шляхом додавання внутрішніх ребер, оскільки від характеру змін, що необхідно внести у роботу моделі, залежить не тільки швидкість, з якою ці зміни будуть виконані, але й швидкодія самої моделі.

### 3. МЕТОД ПЕРЕТВОРЕННЯ GL-МОДЕЛЕЙ ШЛЯХОМ ВВЕДЕННЯ ВНУТРІШНІХ РЕБЕР З УРАХУВАННЯМ ПОПАРНИХ РЕБЕРНИХ ЦИКЛІВ

#### 3.1. Попарні реберні цикли

Зрозуміло, що для блокування втрати зв'язності графа моделі при появі одного з таких векторів, що викликає зникнення в моделі тільки двох ребер, досить проведення одного додаткового ребра зі своєю функцією, яка формується відповідно до [19]. Якщо ж виникає необхідність блокування певної безлічі векторів, то в багатьох випадках також досить одного додаткового ребра (нагадаємо, що ребра в моделі можна переставляти як завгодно). Однак так буває не завжди, зокрема, тоді, коли певні вектори стану утворюють так звані «попарні реберні цикли» (ПРЦ) або більш складні конструкції[20].

Сутність попарних реберних циклів полягає у наступному. Нехай існують 3 вектори стану ВБС, що належать класу  $S$ . Припустимо, що при появі першого з них, з графу GL-моделі видаляються ребра  $i$  та  $j$ , при появі другого –  $j$  та  $k$ , а при появі третього –  $i$  та  $k$ . Реберні функції  $i, j$  та  $k$  утворюють своєрідний цикл, перешкоджаючи блокуванню таких множин за допомогою одного додаткового ребра.

Розглянемо приклад утворення ПРЦ у GL-моделі  $K(2, 7)$ :

Таблиця 3.1

Формування базової моделі  $K(2, 7)$

№ наборів	{1,2,3,4}		{5,6,7}	
1	$K(1,4)$		$K(1,3)$	
	$\{1,2\}$	$\{3,4\}$	$\{5,6\}$	$\{7\}$
2	$K(2,2)$	-	-	-
3	$K(1,2)$	$K(1,2)$	-	-
4	-	$K(2,2)$	-	-

5	-	-	$K(2,2)$	-
6	-	-	$K(1,2)$	$K(1,1)$

На основі таблиці побудуємо базову GL-модель на основі кільцевого графу з 6 ребрами:

$$\begin{aligned}
 f_1 &= x_1 x_2 x_3 x_4 \vee x_5 x_6 x_7 x_8 \\
 f_2 &= x_1 \vee x_2 \\
 f_3 &= x_1 x_2 \vee x_3 x_4 \\
 f_4 &= x_3 \vee x_4 \\
 f_5 &= x_5 \vee x_6 \\
 f_6 &= x_5 x_6 \vee x_7
 \end{aligned}
 \tag{3.1}$$

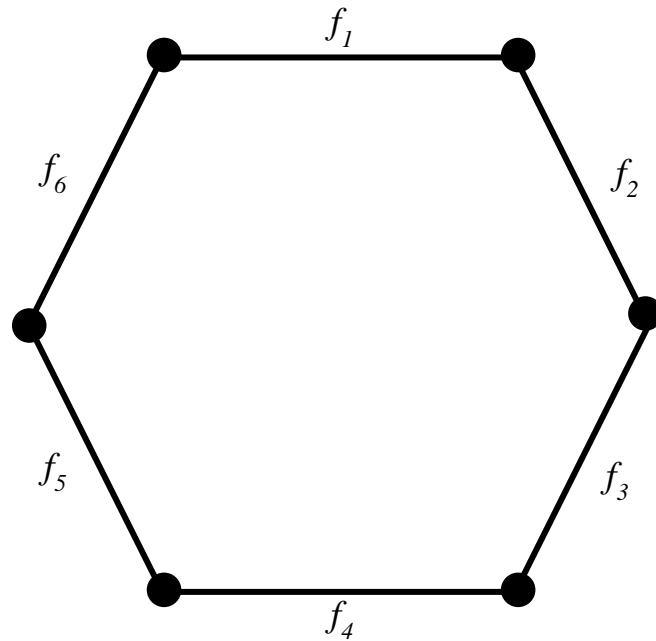


Рис. 3.1. Базова  $K(2,7)$  GL-модель

Припустимо, що множина  $W^+$  складається з трьох векторів:

$$\begin{aligned}
 w_1 &= \{0,0,1,1,0,1,1,1\} \\
 w_2 &= \{0,0,0,1,1,1,1,1\} \\
 w_3 &= \{0,1,0,1,0,1,1,1\}
 \end{aligned}
 \tag{3.2}$$

Розглянемо конфігурації графу, що виникають за кожного з зазначених вище векторів стану:

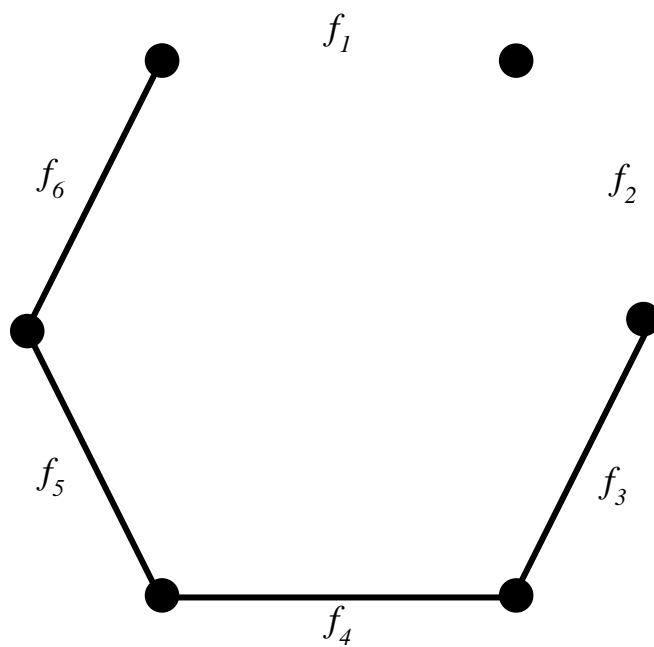


Рис. 3.2. Стан  $K(2,7)$  GL-моделі для вектору  $w_1$

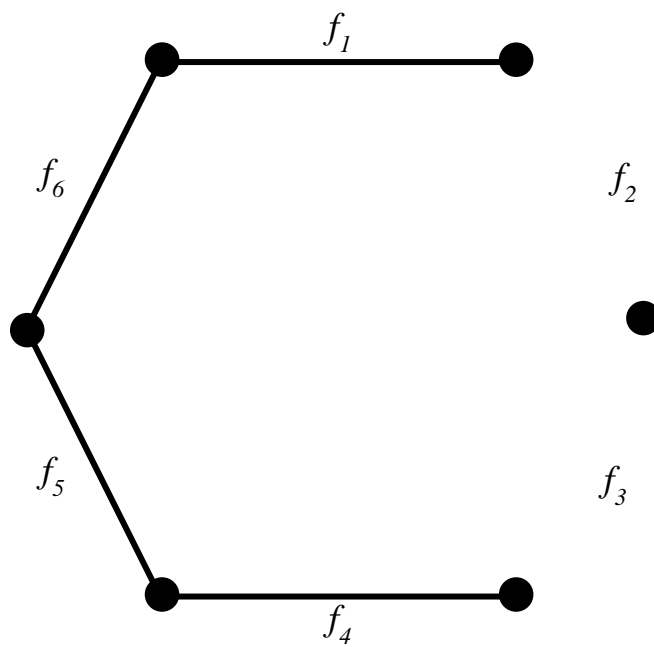


Рис. 3.3. Стан  $K(2,7)$  GL-моделі для вектору  $w_2$

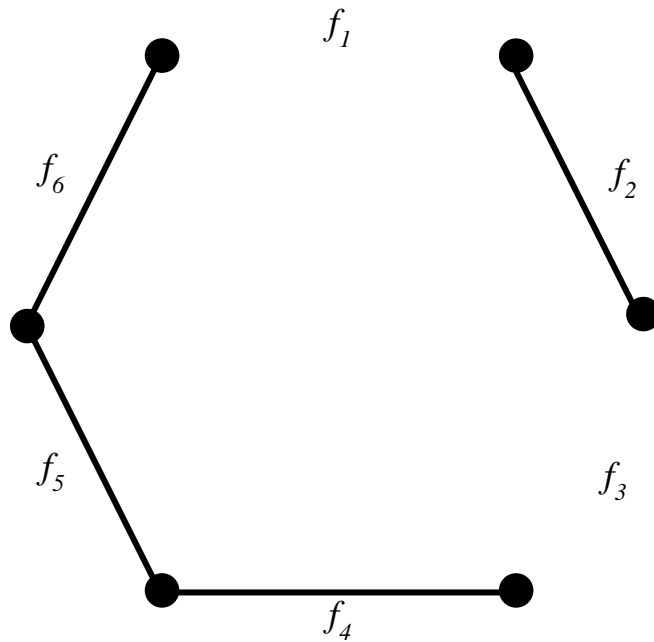


Рис. 3.4. Стан  $K(2,7)$  GL-моделі для вектору  $w_3$

Можна побачити, що пари ребер  $\{f_1, f_2\}$ ,  $\{f_2, f_3\}$  та  $\{f_3, f_1\}$  формують ПРЦ у базовому графі.

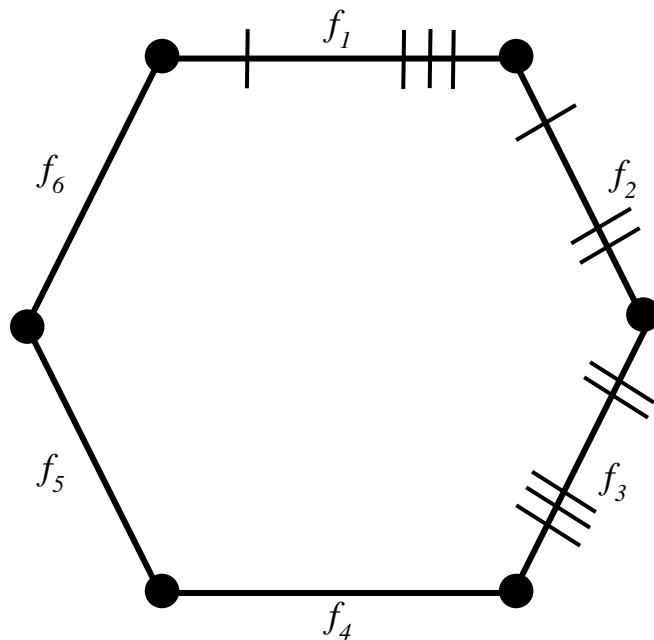


Рис. 3.5. ПРЦ у базовій  $K(2,7)$  GL-моделі

Утворення ПРЦ призводить до того, що стає неможливим блокувати вектори відмов за допомогою лише одного ребра, тому процес створення моделі ускладнюється. Це зумовлене тим, що V-граф, побудований на основі даного прикладу, буде мати цикл, і тому для його розфарбування необхідно буде



щонайменше 3 кольори (що позначає, що граф розділений на щонайменше 3 стійкі S-підмножини).

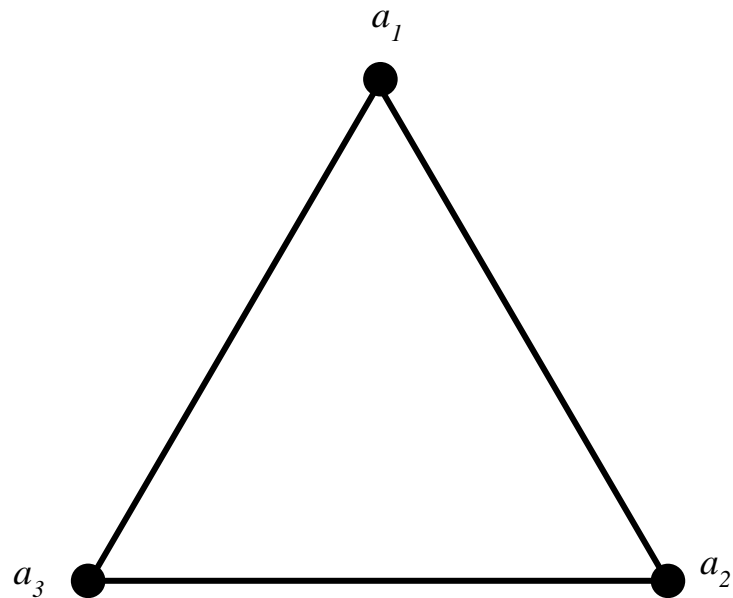


Рис. 3.6. ПРЦ у V-графі базової K(2,7) GL-моделі

Тоді для блокування векторів буде необхідно щонайменше 2 внутрішніх ребра:

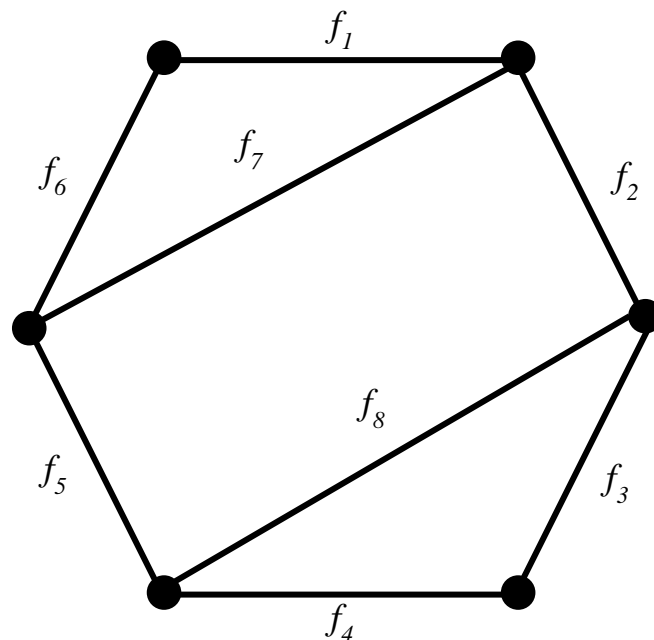


Рис. 3.7. Модифікована K(2,7) GL-модель

$$\begin{aligned} f_1 &= x_1 x_2 x_3 x_4 \vee x_5 x_6 x_7 x_8 \\ f_2 &= x_1 \vee x_2 \end{aligned} \quad (3.3)$$

$$f_3 = x_1 x_2 \vee x_3 x_4$$

$$f_4 = x_3 \vee x_4$$

$$f_5 = x_5 \vee x_6$$

$$f_6 = x_5 x_6 \vee x_7$$

$$f_7 = \bar{x}_1 \bar{x}_2 x_3 x_4 \bar{x}_5 x_6 x_7 x_8 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \bar{x}_5 x_6 x_7 x_8$$

$$f_8 = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 x_5 x_6 x_7 x_8$$

Зрозуміло, що більша кількість додаткових ребер та довший час, що витрачається на зміну структури графу, зменшує ефективність моделі для обчислення стану системи. Тому, ефективна обробка ПРЦ методами модифікації базових GL-моделей, що мають у своїй основі додавання внутрішніх ребер, є надзвичайно важливою.

### **3.2. Відображення підвищення надійності відмовостійких багатопроцесорних систем на GL-моделях з утворенням попарних реберних циклів**

Перетворення моделі шляхом введення додаткових ребер збільшує складність моделі і ускладнює процедуру визначення зв'язності графа. Тому при виборі тієї або іншої моделі поведінки для проектованої системи одним з критеріїв стає наявність ПРЦ, їх кількість і довжина.

Позначимо через  $Q$  множину всіх ПРЦ моделі. Кожен елемент цієї множини є послідовність ребер, що утворюють цикл:

$$\begin{aligned} q_i \in Q, q_i &= \{a_f, a_j, \dots, a_k\} \\ a_f, a_j, \dots, a_k &\in A \end{aligned} \tag{3.4}$$

де  $A$  – це множина усіх ребер графу GL-моделі. Також позначимо  $M$  множину розмірів кожного з  $q_i$ .

Розглянемо окремі випадки структури графу  $V$ , за якої утворюється ПРЦ. Для цього будемо використовувати приклад  $K(2,9)$  базової GL-моделі, що необхідно модифікувати:

Таблиця 3.2

Формування базової моделі  $K(2, 9)$

№ наборів	{1,2,3,4}		{5,6,7,8,9}		
1	$K(1,4)$		$K(1,5)$		
	{1,2}	{3,4}	{5,6}	{7, 8}	{9}
2	$K(2,2)$	-	-	-	-
3	$K(1,2)$	$K(1,2)$	-	-	-
4	-	$K(2,2)$	-	-	-
5	-	-	$K(2,2)$	-	-
6	-	-	$K(1,2)$	$K(1,3)$	
7	-	-	-	$K(2,2)$	-
8				$K(1,2)$	$K(1,1)$

На основі таблиці побудуємо базову GL-модель на основі кільцевого графу з 8 ребрами:

$$\begin{aligned}
 f_1 &= x_1 x_2 x_3 x_4 \vee x_5 x_6 x_7 x_8 x_9 \\
 f_2 &= x_1 \vee x_2 \\
 f_3 &= x_1 x_2 \vee x_3 x_4 \\
 f_4 &= x_3 \vee x_4 \\
 f_5 &= x_5 \vee x_6 \\
 f_6 &= x_5 x_6 \vee x_7 x_8 x_9 \\
 f_7 &= x_7 \vee x_8 \\
 f_8 &= x_7 x_8 \vee x_9
 \end{aligned}
 \tag{3.5}$$

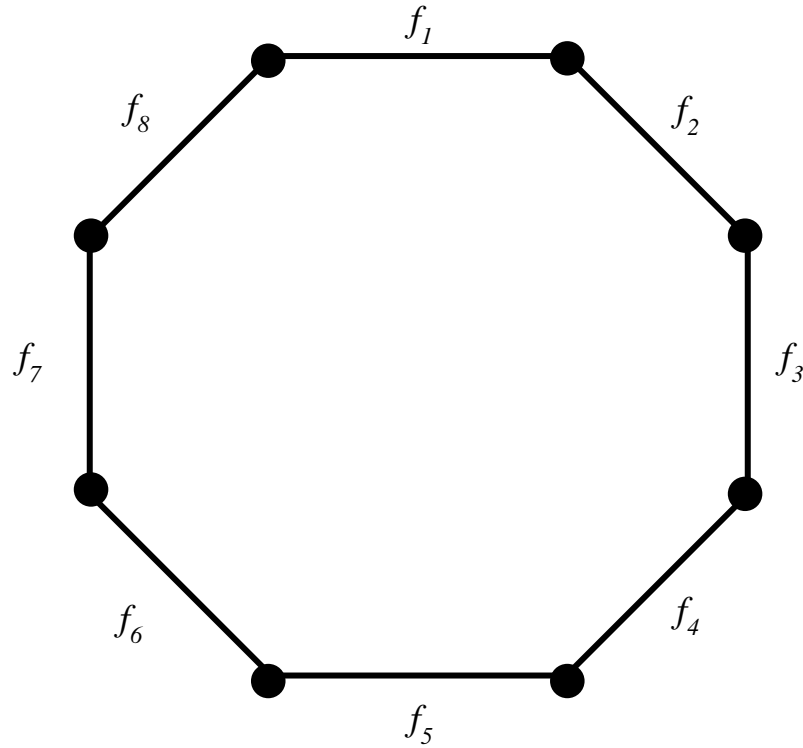


Рис. 3.8. Базова K(2,9) GL-модель

Позначимо  $V'$  граф, як підграф  $V$ -графа, множина вершин якого утворюється на базі  $m(Q')$ , де  $Q'$  – певна підмножина  $Q$ ,  $p'$  – хроматичне число  $V'$ . На  $Q'$  накладається додаткове обмеження:

$$m(q_i) \notin m(Q'), \forall q_i \in Q/Q' \quad (3.6)$$

Спочатку проведемо первісну оцінку максимальної кількості ребер, яку може потребувати ПРЦ для модифікації моделі. Для цього розглянемо випадок, коли  $Q' \subseteq Q$ . Нехай  $q_{max} = \max(|q_i|), \forall q_i \in Q'$ .

ПРЦ являє собою елементарний цикл у  $V$ -графі. Виділивши у  $V'$ -графі цикл найбільшої довжини, можна сказати, що хроматичне число  $V'$ -графа не буде більше кількості вершин цього графу (та буде дорівнювати йому за випадку, коли  $V'$ -граф – повний). З цього витікає, що:

$$p' \leq q_{max} \quad (3.7)$$

Згідно наведених у розділі 2 залежностей хроматичного числа  $V$ -графа та кількості необхідних для модифікації моделі ребер, можна вивести наступну формулу:

$$l \leq \left\lfloor \frac{q_{max}}{2} \right\rfloor \quad (3.8)$$

Розглянемо випадок, коли  $Q$  містить лише цикли парної довжини. Для цього будемо використовувати наступні вектори, відмову модулів у яких необхідно заблокувати:

$$\begin{aligned}
 w_1 &= \{0,0,1,1,0,1,1,1,1\} \\
 w_2 &= \{0,0,0,1,1,1,1,1,1\} \\
 w_3 &= \{0,1,0,0,1,1,1,1,1\} \\
 w_4 &= \{1,1,0,0,0,1,1,1,1\} \\
 w_5 &= \{0,1,1,1,0,0,1,1,1\} \\
 w_6 &= \{1,1,1,1,0,0,0,1,1\} \\
 w_7 &= \{1,1,1,1,0,1,0,0,1\} \\
 w_8 &= \{0,1,1,1,1,1,0,0,1\}
 \end{aligned}
 \tag{3.9}$$

Згідно теоремі Кьонінга, граф можна розфарбувати двома кольорами тоді і тільки тоді, коли у ньому є лише цикли парної довжини. Згідно цьому твердженню, можна розфарбувати  $V'$ -граф наведеного вище прикладу 2 кольорами. Тоді достатньо лише одного ребра для блокування усіх векторів, що призводять до утворення такої структури  $V'$ -графу.

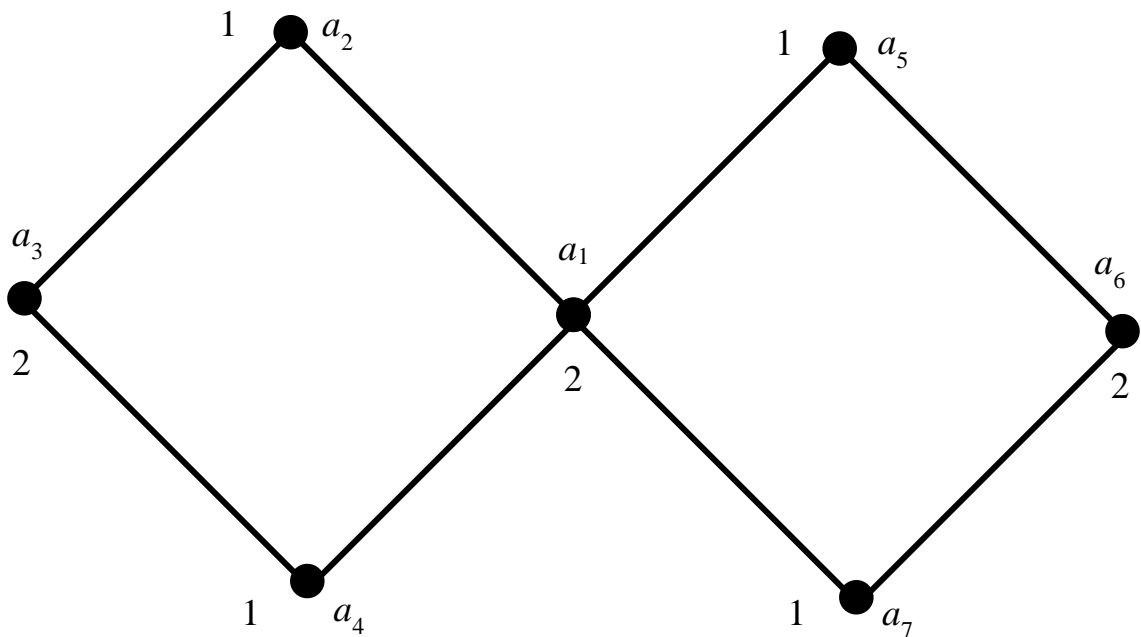


Рис. 3.9.  $V'$ -граф для векторів 3.9

Після розфарбування графу, пошук місця проведення додаткового ребра проводиться за методом, зазначеним у розділі 2.

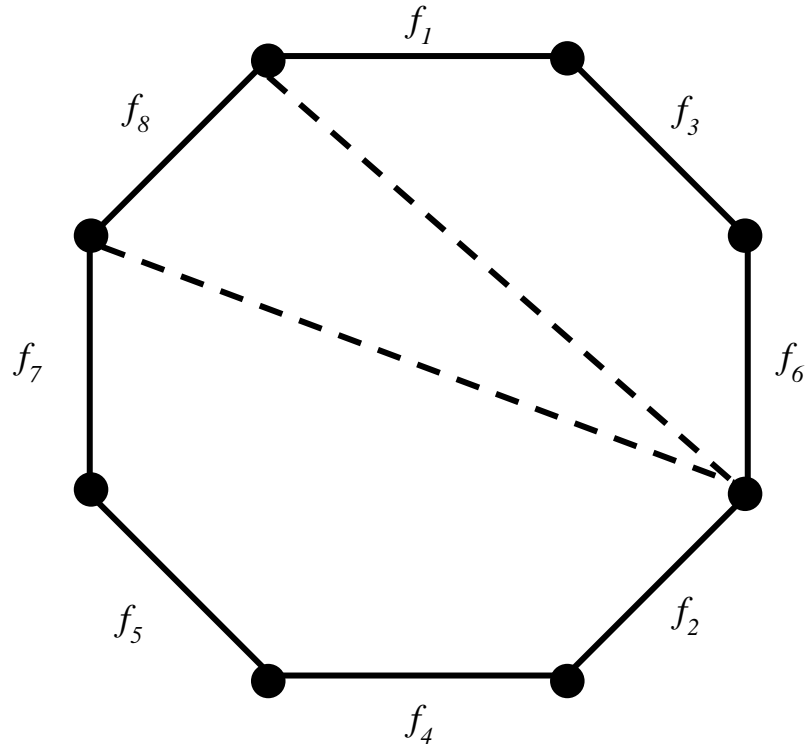


Рис. 3.10. Можливі місця проведення додаткового ребра для векторів 3.9

Таким чином, ПРЦ, що складаються лише з парної кількості ребер, не зашкоджують швидкодії моделі, оскільки вони потребують лише одного додаткового ребра для блокування. Одне й те саме ребро може бути використане для блокування декількох таких ПРЦ.

Розглянемо тепер  $Q'$ , що містить лише цикли непарної довжини. Для цього будемо використовувати наступні вектори, відмову модулів у яких необхідно заблокувати:

$$\begin{aligned}
 w_1 &= \{0,0,1,1,0,1,1,1,1\} \\
 w_2 &= \{0,0,0,1,1,1,1,1,1\} \\
 w_3 &= \{0,1,0,1,0,1,1,1,1\} \\
 w_4 &= \{0,1,1,1,0,0,1,1,1\} \\
 w_5 &= \{1,1,1,1,0,0,0,1,1\} \\
 w_6 &= \{0,1,1,1,1,1,0,0,1\}
 \end{aligned}
 \tag{3.10}$$

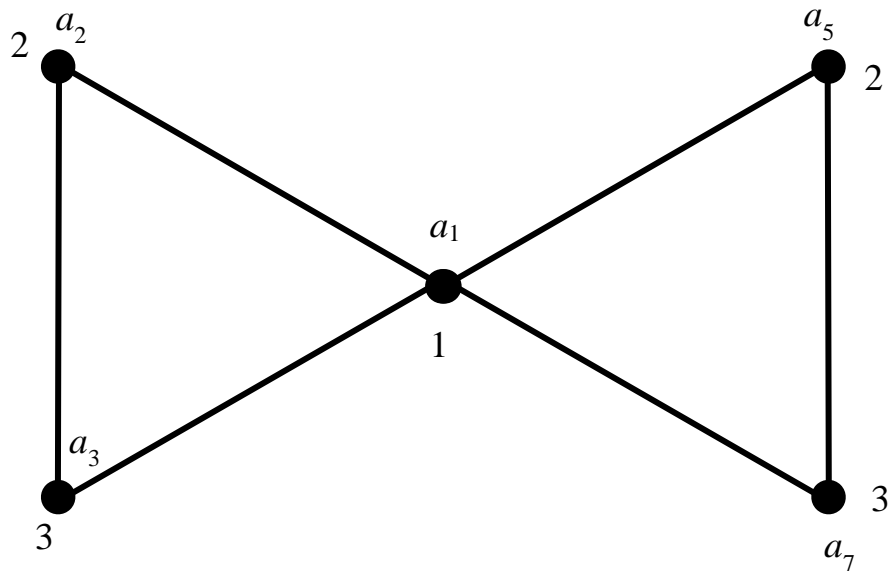


Рис. 3.11.  $V'$ -граф для векторів 3.10

Для такого графу перетин будь-яких двох циклів не може мати більше ніж одну вершину. Оскільки для розфарбування кожного такого циклу необхідно 3 кольори, то й сам  $V'$ -граф потребує 3 кольори для розфарбування. Зрозуміло, що маючи у такому разі хроматичне число графа  $p'=3$  можна визначити кількість необхідних для блокування ребер, що дорівнює 2.

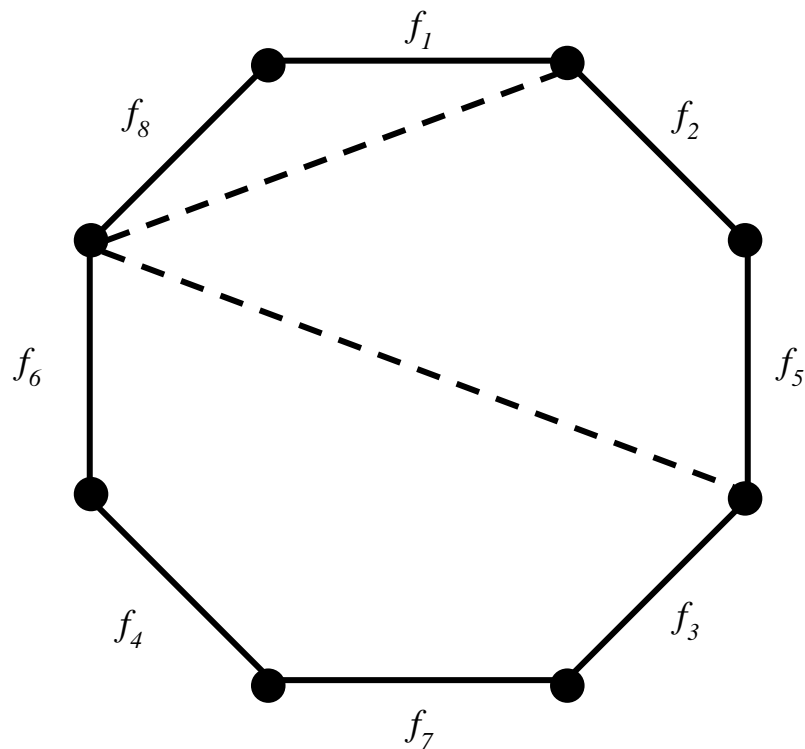


Рис. 3.12. Можливі місця проведення додаткових ребер для векторів 3.10

Зрозуміло, що поява ПРЦ такої конфігурації призводить до збільшення часу, необхідного на те, щоб визначити стан системи в залежності від стану її модулів за допомогою модифікованої GL-моделі.

Розглянемо комбінації зазначених вище типів ПРЦ. Нехай існує ребро  $a_x$ , таке, що:

$$a_x = \cup (m(q_i) \cap m(q_j)), \forall q_i, q_j \in Q', i \neq j \quad (3.11)$$

тобто  $a_x$  – єдине спільне ребро для всіх ПРЦ у  $Q'$ .

Спочатку розглянемо випадок, коли хоча б один з ПРЦ у  $Q'$  непарної довжини (вектори з 3.10 відповідають цій умові). Наведене вище визначення  $a_x$  означає, що  $V'$ -граф має лише одну спільну вершину для усіх ПРЦ, тому для того, щоб розфарбувати весь граф необхідно не більше кольорів, ніж на розфарбування будь-якого з них окремо. Як ми вже визначили, графи непарної довжини потребують 3 кольори для розфарбування, тому кількість кольорів для розфарбування  $V'$ -граф також дорівнює 3.

Випадок, коли усі ПРЦ з  $Q'$  з одним спільним ребром мають парну довжину, є тривіальним. У такому разі розфарбування  $V'$ -графу нічим не відрізняється від рофарбування звичайного  $V'$ -графу з ПРЦ парної довжини, тому для блокування такого  $Q'$  необхідно лише одно ребро, як вже було показано вище. Приклад такої модифікації було показано на векторах 3.9.

Таким чином, вплив ПРЦ на складність модифікації та подальшої експлуатації модифікованої моделі залежить від того, чи є ПРЦ у  $Q$  непарної довжини. Можна зазначити, що наявність ПРЦ непарної довжини, що утворюються під час перетворення базових GL-моделей може бути важливим індикатором складності структури моделі, що буде отримана в результаті модифікації.

Нехай існує лише одна пара ребер, на яких ПРЦ з  $Q'$  перетинаються:

$$\{a_x, a_y\} = \cup (m(q_i) \cap m(q_j)), \forall q_i, q_j \in Q', i \neq j \quad (3.12)$$

Розглянемо таку систему на прикладі:

$$\begin{aligned} w_1 &= \{0,0,1,1,0,1,1,1,1\} \\ w_2 &= \{0,0,0,1,1,1,1,1,1\} \end{aligned} \quad (3.13)$$



$$w_3 = \{0,1,1,1,0,0,1,1,1\}$$

$$w_4 = \{1,1,1,1,0,0,0,1,1\}$$

$$w_5 = \{1,1,1,1,0,1,0,0,1\}$$

$$w_6 = \{0,1,1,1,1,1,0,0,1\}$$

$$w_7 = \{1,1,1,1,0,1,0,1,0\}$$

$$w_8 = \{1,1,0,1,1,0,0,1,1\}$$

Зрозуміло, що для розфарбування всього  $V'$ -графу у такому разі потребується не більше кольорів, ніж на розфарбування його циклів окремо. Це, в свою чергу, визначає кількість кольорів, необхідну для розфарбування  $V'$ -графу: 2 для випадку, коли хоча б один з циклів має непарну довжину та 1, якщо всі цикли мають парну довжину.

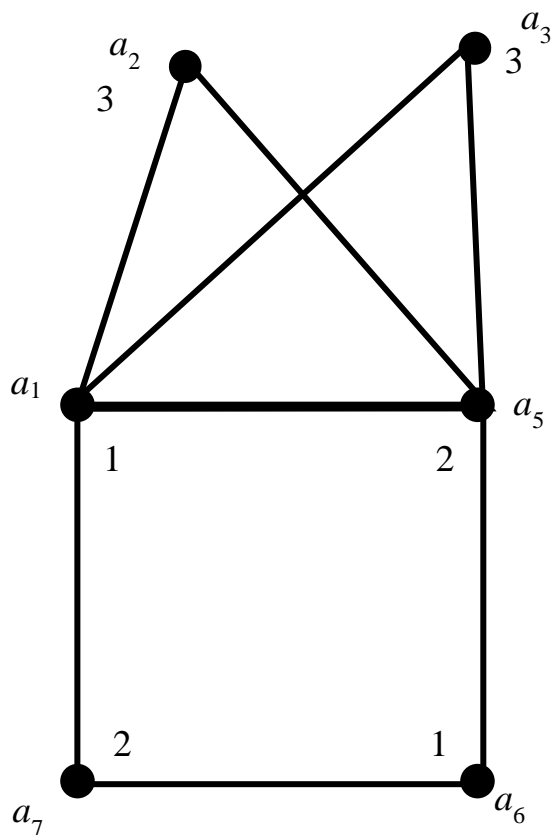


Рис. 3.13.  $V'$ -граф для векторів 3.13

Обчисливши хроматичне число графа, можна перебачити необхідні додаткові внутрішні ребра для модифікації:

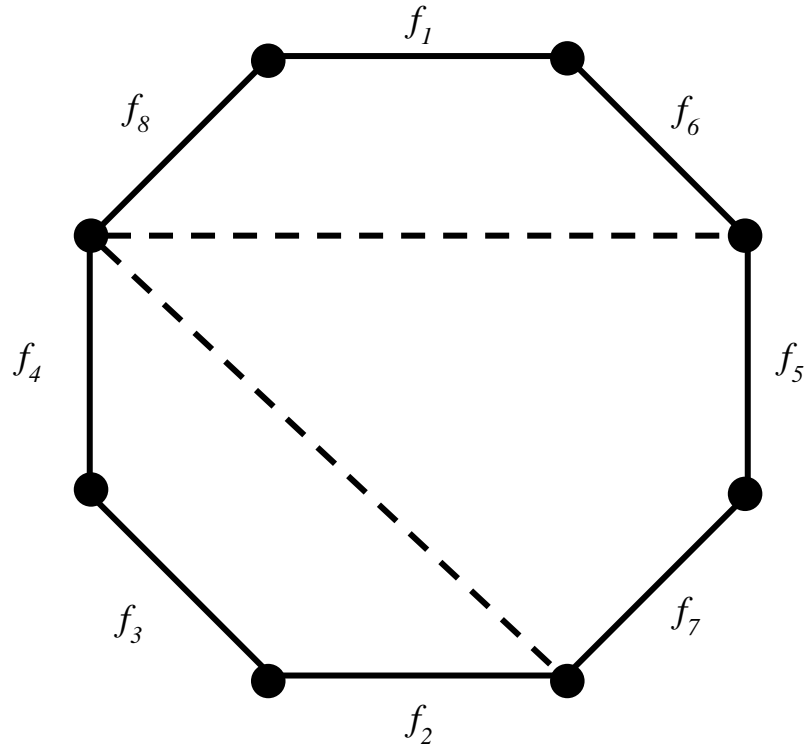


Рис. 3.14. Можливі місця проведення додаткових ребер для векторів 3.13

Розглянемо загальний випадок перетинання ребер ПРЦ у  $Q'$ . Нехай існує така множина ребер  $A'$ , що:

$$\begin{aligned} A' &\subseteq A, |A'| \geq 3 \\ A' &\supseteq m(q_i \cap m(q_j)), \forall q_i, q_j \in Q', i \neq j \end{aligned} \quad (3.14)$$

$A'$  у цьому разі є множиною усіх ребер ПРЦ, що перетинаються. Розглянемо таку структуру ПРЦ на прикладі:

$$\begin{aligned} w_1 &= \{0,0,1,1,0,1,1,1,1\} \\ w_2 &= \{0,0,0,1,1,1,1,1,1\} \\ w_3 &= \{0,1,0,1,0,1,1,1,1\} \\ w_4 &= \{0,1,1,1,0,0,1,1,1\} \\ w_5 &= \{1,1,1,1,0,0,0,1,1\} \\ w_6 &= \{1,1,1,1,0,1,0,0,1\} \\ w_7 &= \{0,1,1,1,1,1,0,0,1\} \\ w_8 &= \{1,1,1,1,0,1,0,1,0\} \\ w_9 &= \{1,1,1,1,1,1,0,0,0\} \end{aligned} \quad (3.15)$$

Попарний перетин усіх ПРЦ входить до підмножини  $A'$  та на розфарбування цієї підмножини потрібно не більше кольорів, ніж потужність  $A'$ . Для розфарбування всіх інших вузлів графу  $V'$  необхідна менша кількість кольорів. Таким чином, кількість додаткових ребер, необхідних для блокування такого  $Q'$  не буде вищою, ніж  $\left\lfloor \frac{|A'|}{2} \right\rfloor$ . Таким чином, чим більше точок перетину мають ПРЦ, тим складніше буде модель, утворена після блокування  $Q'$  за допомогою методу перетворення базових моделей на небазові шляхом додавання внутрішніх ребер.

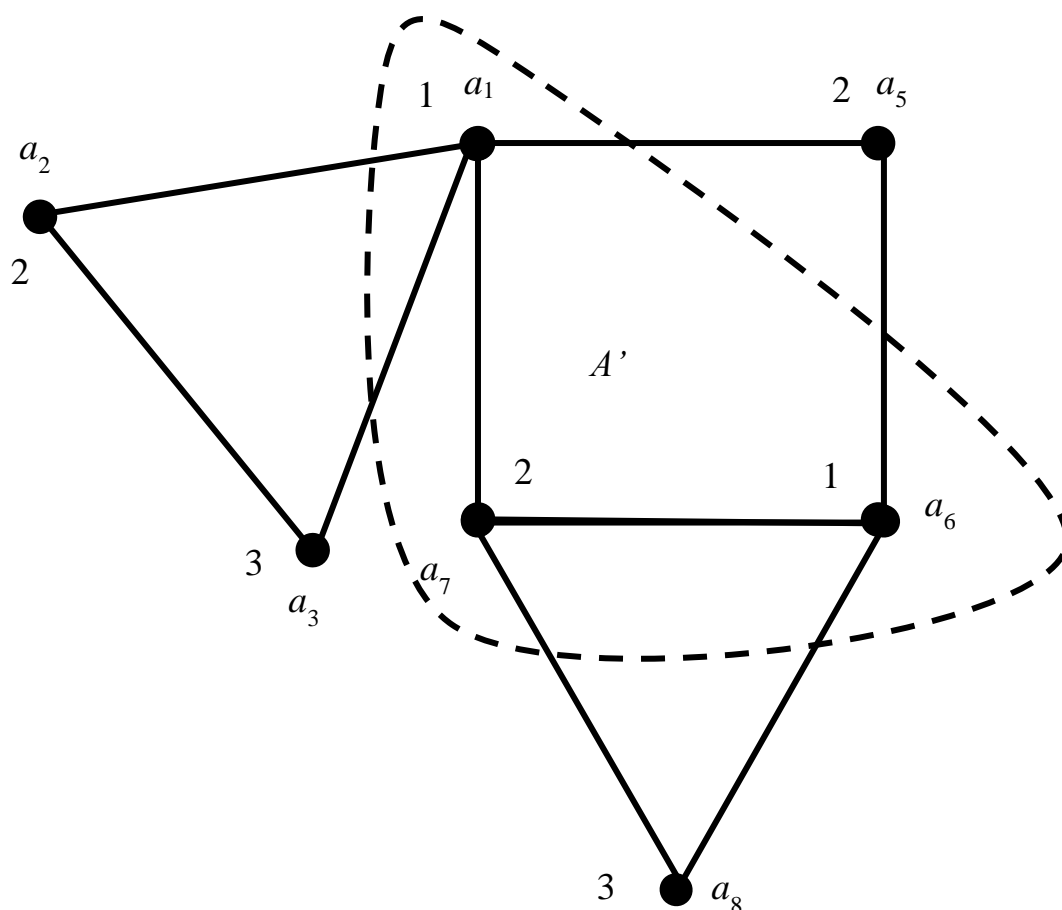


Рис. 3.15.  $V'$ -граф для векторів 3.15

Визначивши кольори вершин  $V'$ -можна визначити необхідні для блокування векторів 3.15 ребра.

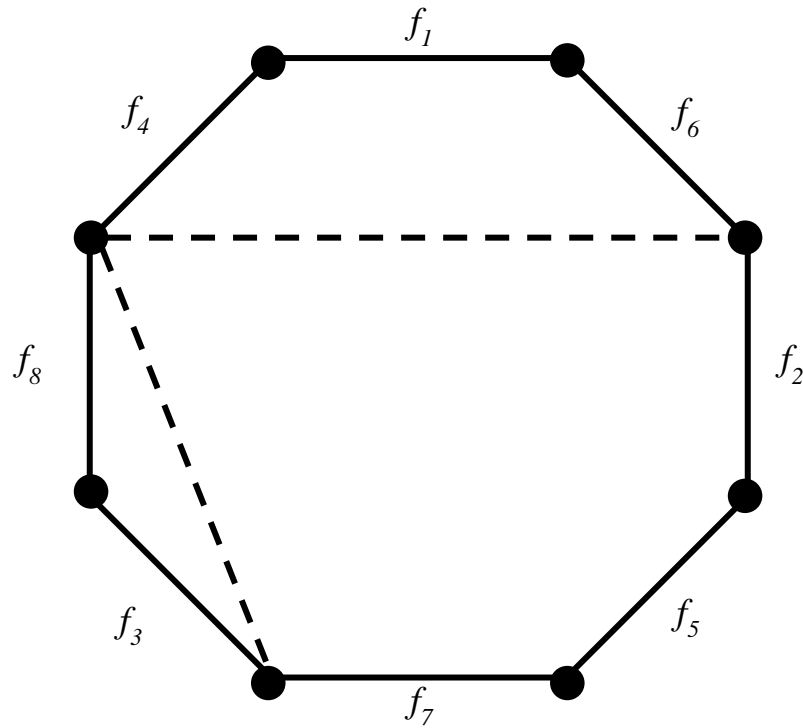


Рис. 3.16. Можливі місця проведення додаткових ребер для векторів 3.15

Додаткові ребра суттєво ускладнюють модель, і, відповідно, збільшують час, необхідний для розрахунку надійності системи з необхідною точністю.

А оскільки для створення небазової системи можна використовувати різні методи, то практичний інтерес викликає попередній аналіз можливих шляхів отримання небазових моделей. Метод перетворення базових моделей шляхом додавання внутрішніх ребер може у залежності від характеристик системи, що моделюється суттєво ускладнити модель, тим самим знижуючи її ефективність.

Методи, що враховують ПРЦ у процесі модифікації, можуть бути використані для того, щоб зменшити цей негативний ефект. Окрім того, розглянуті оцінки кількості необхідних для перетворення додаткових внутрішніх ребер можуть бути використані на етапі планування перетворення. Якщо автоматична система, яка займається побудовою GL-моделей за відомими характеристиками, визначає, що певна призначена для моделювання ВБС буде містити велику кількість ПРЦ, тоді інші методи перетворення базових GL-моделей у небазові можуть набути більшого пріоритету.

Також перевіряти таким чином можна не тільки наявність у системі, що моделюється, ПРЦ, але й визначати кількість ребер у перетинанні циклів та характер перетинання, кількість ребер у окремих циклах, тощо.

### **Висновки розділу 3**

В розділі проаналізовано явище ПРЦ та наслідки їх появи при перетворенні базових GL-моделей у небазові. Розроблений метод перетворення базових GL-моделей у небазові з урахуванням ПРЦ. Розглянуті різноманітні конфігурації ПРЦ та їх вплив на роботу метода. Наведений метод дозволяє обчислити мінімальну кількість необхідних для блокування ПРЦ додаткових ребер, а також може бути застосований для оцінки ефективності отриманих моделей.

## ВИСНОВОК

В дисертації розглянуто проблему використання моделей для відображення стану ВБС в залежності від стану їх модулів та обчислення ймовірності безвідмовної роботи статистичним методом шляхом проведення експериментів, використовуючи зазначені моделі. Проаналізовані існуючі рішення для представлення відмовостійких систем та зокрема ВБС, розглянуто принципи їх роботи, переваги та недоліки.

При цьому показано, що GL-моделі, поєднуючи властивості графів та булевих функцій, дозволяють моделювати системи з високою ефективністю, та легко масштабуються, дозволяючи обчислювати стан системи з великою кількістю модулів швидше, ніж інші розглянуті типи моделей.

В роботі аналізуються методи формування GL-моделей, розглядається проблема перетворення базових моделей в небазові. Проводиться аналіз існуючих методів модифікації GL-моделей. Визначено, що метод, заснований на додаванні внутрішніх ребер для забезпечення зв'язності графу GL-моделі є більш гнучким для створення небазових моделей, тому запропонований метод було вирішено засновувати на ньому.

Аналіз існуючих методів показав, що швидкодія отриманих після перетворення моделей залежить від множини векторів стану, за яких необхідно блокувати втрату зв'язності GL-моделі, та їх взаємних залежностей. Встановлено, що певні набори векторів утворюють так звані ПРЦ, які негативно впливають на швидкодію отриманих моделей. У дисертації аналізуються різноманітні конфігурації ПРЦ та пропонуються способи їх блокування.

В процесі виконання роботи отримано наступні результати:

1. Розроблено метод перетворення базових моделей у небазові з урахуванням ПРЦ.

2. Розроблено метод оцінки мінімальної кількості додаткових внутрішніх ребер, необхідних для блокування втрати зв'язності графу GL-моделі з ПРЦ.



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Надежность и эффективность в технике. Справочник в 10 т. (Ред. совет: В. С. Авдуевский (пред.) и др. Т. 1. Методология. Организация. Терминология)/Под ред. А. И. Рембезы. – М.: Машиностроение, 1989. – 224 с.
2. Data Processing Vocabulary. Section 14. Reliability, Maintenance and Availability. – Geneva: ISO 2382, 1976. – 16 p.
3. Шкляр В. Н. Надежность систем управления: учебное пособие / В. Н. Шкляр; Томский политехнический университет. – Томск: Издательство Томского политехнического университета, 2009. – 126 стр.
4. Павловская, О. О. Основы теории надежности: учебное пособие / О.О. Павловская. – Челябинск: Издательский центр ЮУрГУ, 2013. – 94 с.
5. Федотов, А. В. Основы теории надежности и технической диагностики: конспект лекций / А. В. Федотов, Н. Г. Скабкин. – Омск : Изд-во ОмГТУ, 2010. – 64 с.
6. Коваленко А.Г. Отказоустойчивые микропроцессорные системы / Коваленко А.Г , В.В. Гула – К.: Техніка, 1986 – 148с.
7. Kuo W., Zuo Ming J. The k-out-of-n system model [Електронний ресурс] / Kuo W., Zuo Ming J. – [50 p.]. – Режим доступу: <http://www.ewp.rpi.edu/hartford/~ernesto/S2008/SMRE/Papers/Kuo-Zuokoon.pdf>. – Назва з екрану.
8. Barlow R.E. Computing k-out-of-n system reliability/ R. E. Barlow, K. D. Heidtmann // IEEE Trans. Reliab. – 1984. – Vol. 33. – pp.322-323.
9. Pham H. On the estimation of reliability of k-out-of-n systems/ Pham H. – International Journal of Systems Assurance Engineering and Management, 2010 – vol. 1, no. 1 – pp. 32-35.
10. Endharta A. J. Preventive maintenance policy for linear consecutive k-out-of-n: F system [Електронний ресурс] / A. J. Endharta, W. Y. Yun, Yamamoto H. – [13 p.]. – Режим доступу:

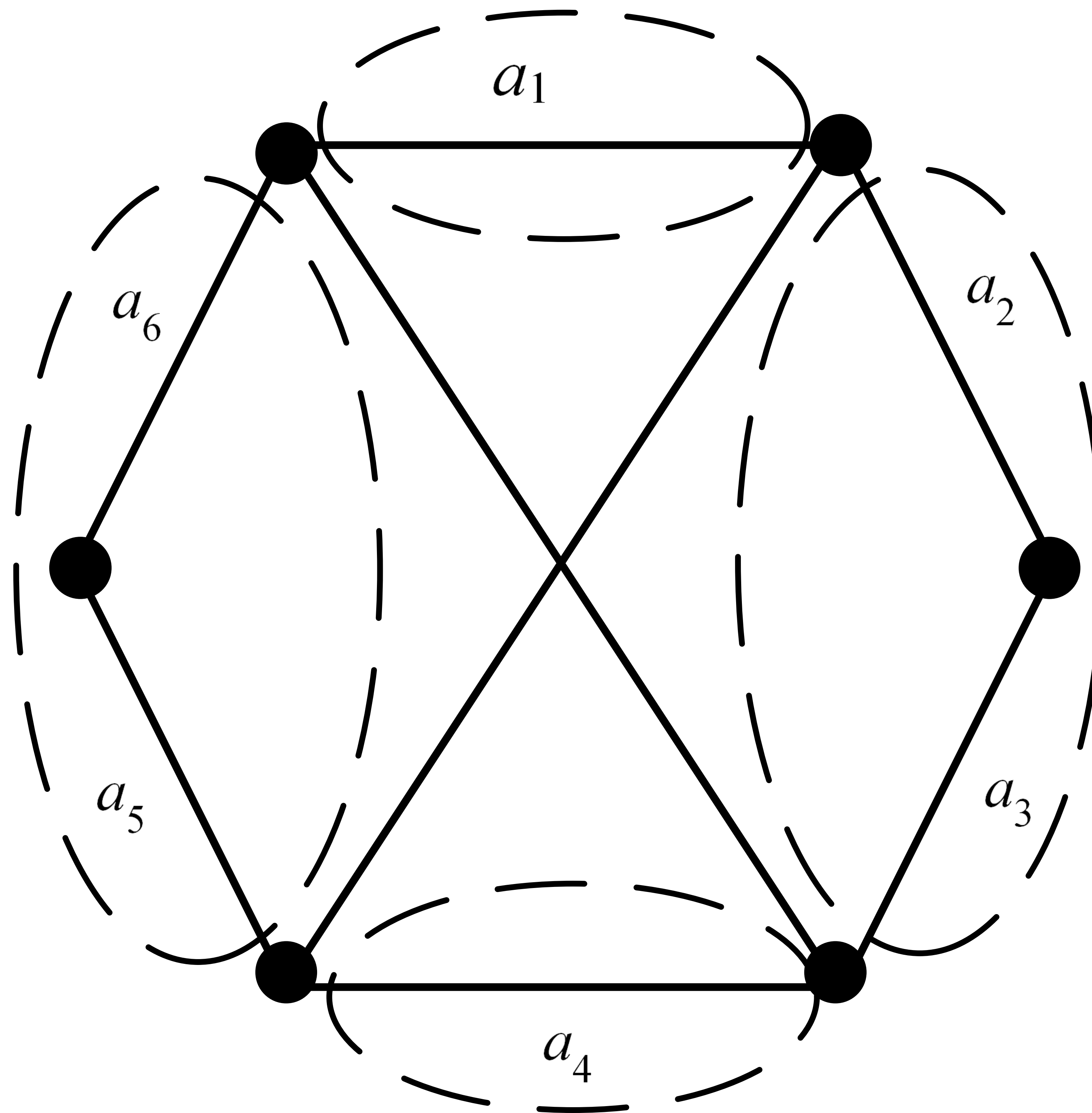
[http://www.orsj.or.jp/~archive/pdf/e\\_mag/Vol.59\\_04\\_334.pdf](http://www.orsj.or.jp/~archive/pdf/e_mag/Vol.59_04_334.pdf). – Назва з екрану.

11. A. E. Gera Combined  $k$ -out-of- $n$ :G, and Consecutive  $k$ -out-of- $n$ :G systems// IEEE Transactions on Reliability, vol. R-53, pp. 523-531, Dec. 2004.
12. Wegener I. The complexity of Boolean functions / Wegener I. – Stuttgart: Teubner, 1987. – 458 p.
13. Райншке К. Оценка надежности систем с использованием графов / К. Райншке, И. А. Ушаков/Под ред. И. А. Ушакова. М.: Радио и связь, 1988. 208с.
14. Самофалов К. Г. Прикладная теория цифровых автоматов / К. Г. Самофалов, А. М. Романкевич, В. Н. Валуйский, Ю. С. Каневский, М. М. Пиневиц – К. :Вища школа, 1987. – 375с.
15. Романкевич А. М. Графо-логические модели для анализа сложных отказоустойчивых вычислительных систем/ Романкевич А. М., Карачун Л. Ф., Романкевич В. А. // Электронное моделирование.– 2001.–т.23, № 1.– С. 102-111.
16. Романкевич О. М. Аналіз відмовостійких багатопроцесорних систем на основі графо-логічних моделей нециклічного типу/ О. М. Романкевич, В. О. Романкевич, О. В. Богуславський, Ал Шбул Рабах. // Вісник Технологічного університету Поділля. – 2002. – т.2, №3. – с. 30-33.
17. Романкевич В. А. Об одной модели поведения отказоустойчивой многопроцессорной системы / В. А. Романкевич // Радиоэлектроника и информатика. – 1999. – №1. – С.75-76.
18. Романкевич О. М. До питання побудови моделі поведінки багатомодульних систем / О. М. Романкевич, Л. Ф. Карачун, В. О. Романкевич // Наукові вісті НТУУ “КПІ”. – 1998. – №1. – с. 38-40.
19. Романкевич В. А. Об одном способе построения GL-моделей отказоустойчивых многопроцессорных систем / В. А. Романкевич //

Радіоелектронні і комп'ютерні системи.–2006.– № 7.– С. 47-51.

20. Романкевич А. М. О сложности преобразования GL-моделей на ранних этапах проектирования отказоустойчивых многопроцессорных систем / А. М. Романкевич, И. В. Майданюк, Е. Р. Потапова // Радіоелектронні і комп'ютерні системи. - 2009. - № 7. - С. 74–77.

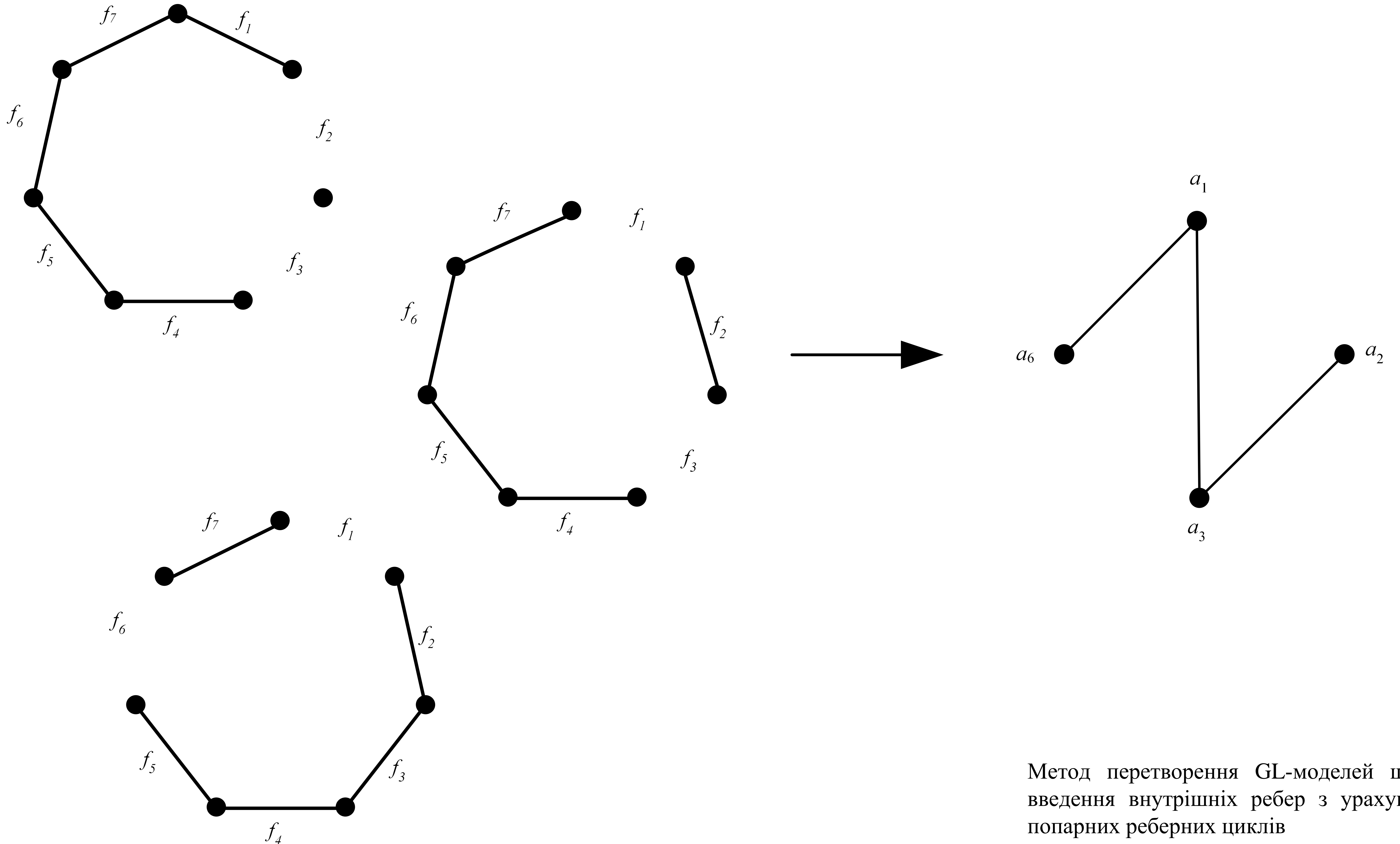
# Стабільні S-підмножини



□

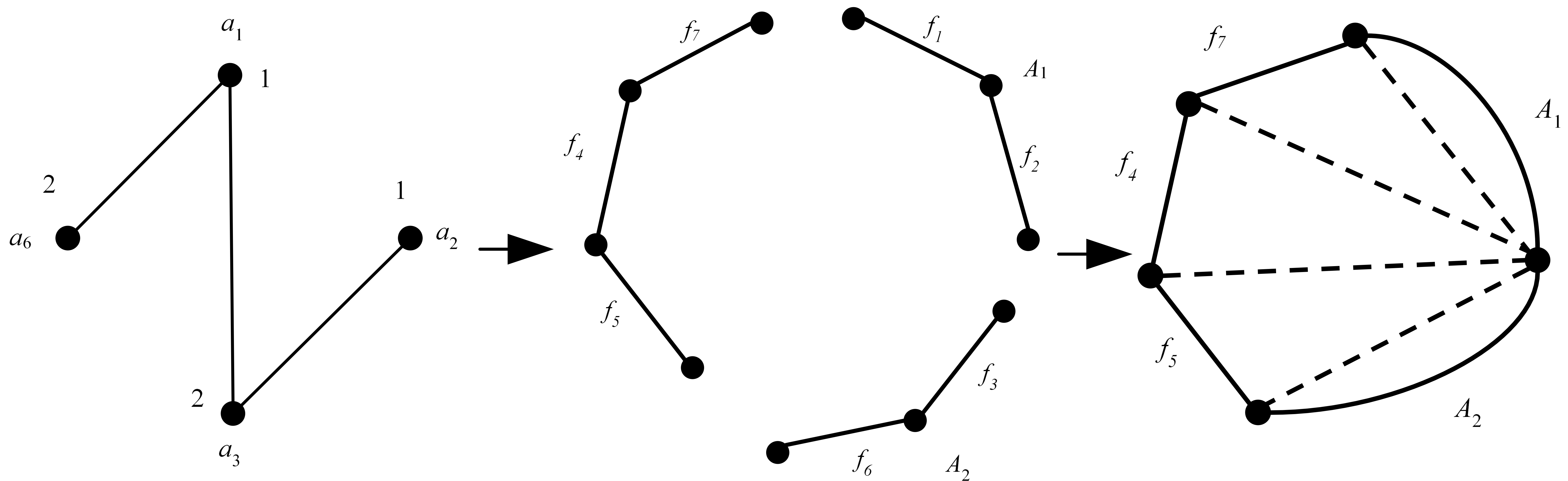
Метод перетворення GL-моделей шляхом  
введення внутрішніх ребер з урахуванням  
попарних реберних циклів  
Примак І.К., група КВ-62М

# Побудова V-графу



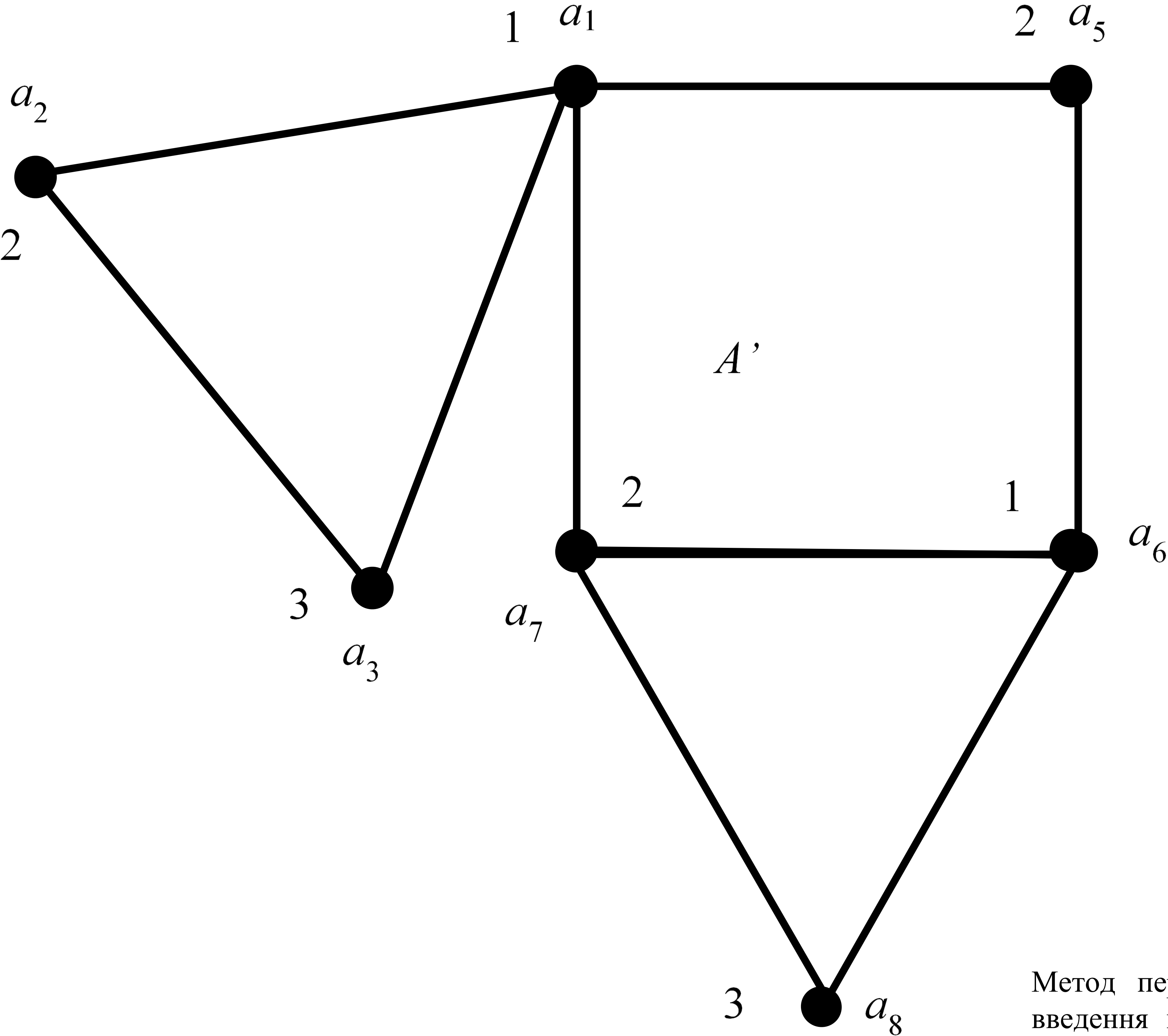
Метод перетворення GL-моделей шляхом введення внутрішніх ребер з урахуванням попарних реберних циклів  
 Примак І.К., група KB-62М

# Побудова модифікованого графу



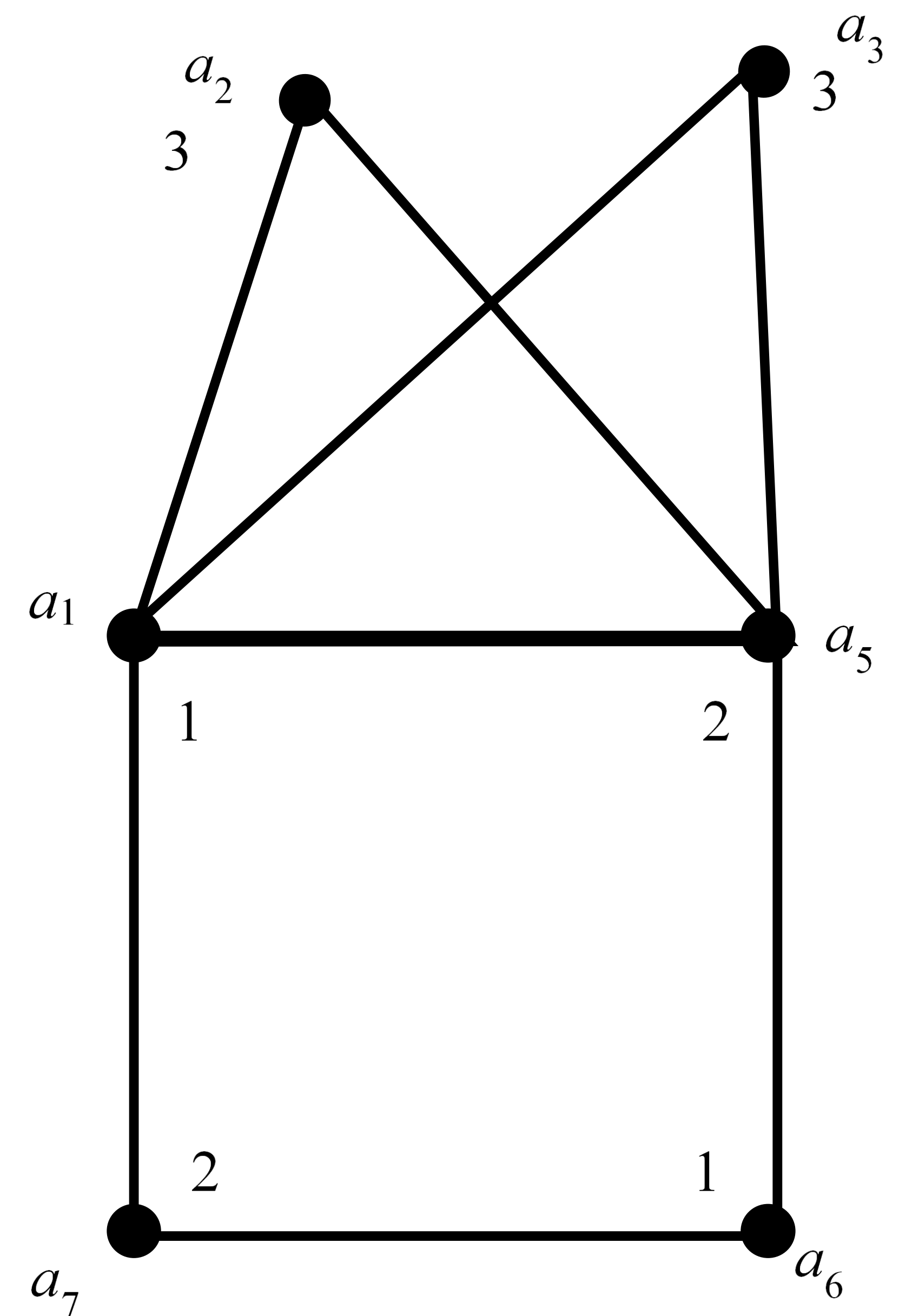
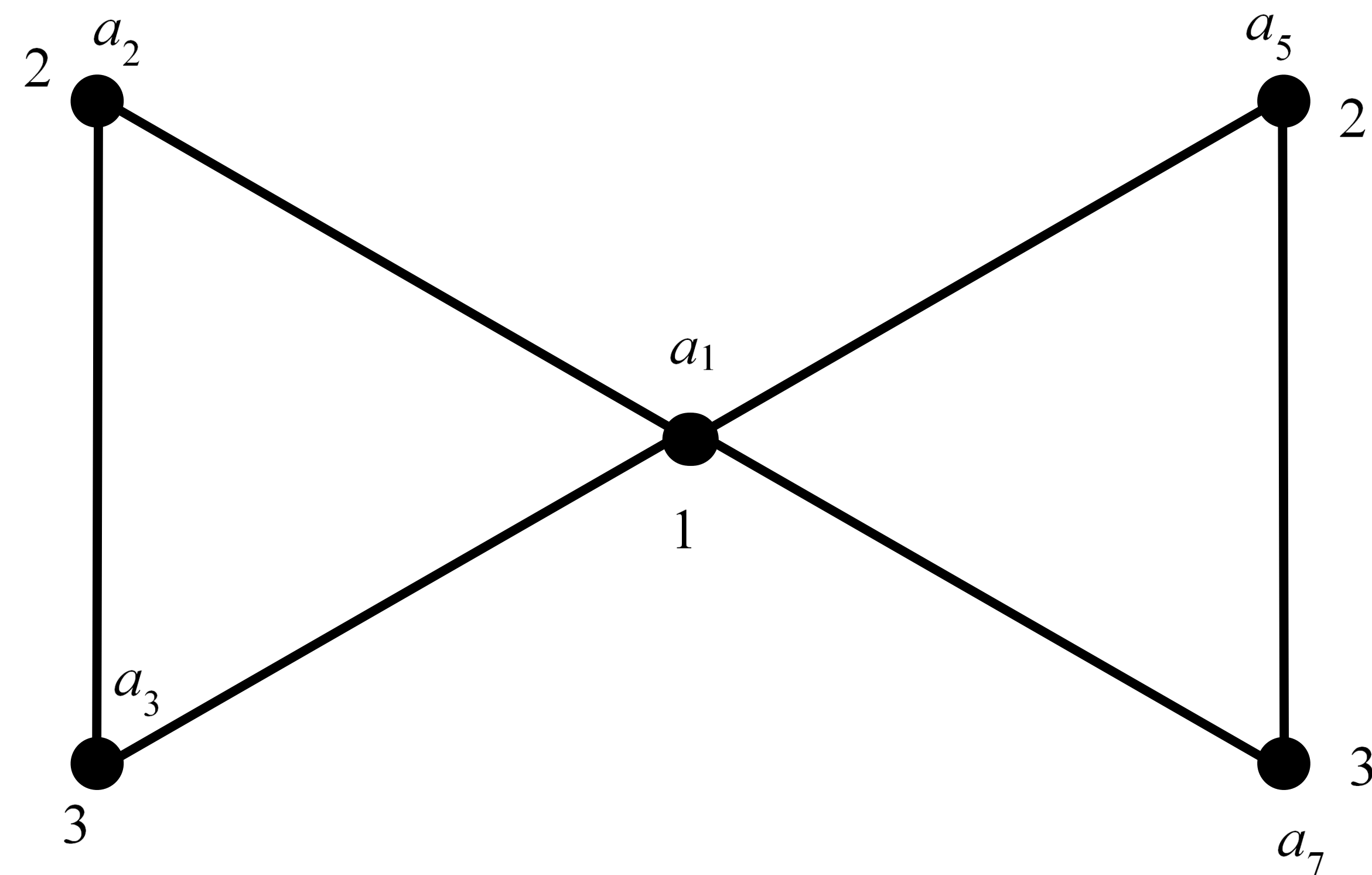
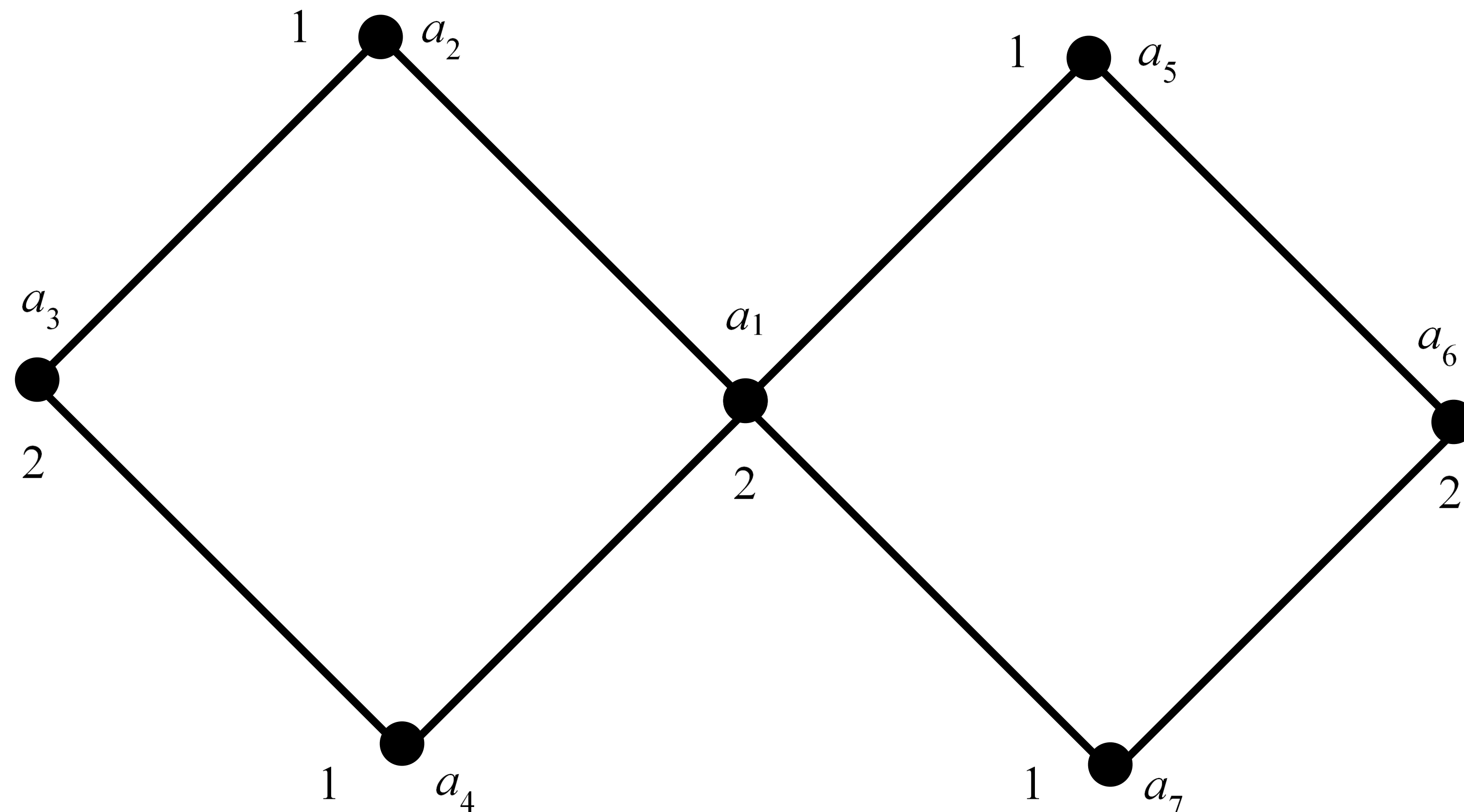
Метод перетворення GL-моделей шляхом  
введення внутрішніх ребер з урахуванням  
попарних реберних циклів  
Примак І.К., група КВ-62М

$V'$ -граф для моделі з ПРЦ



Метод перетворення GL-моделей шляхом  
введення внутрішніх ребер з урахуванням  
попарних реберних циклів  
Примак І.К., група КВ-62М

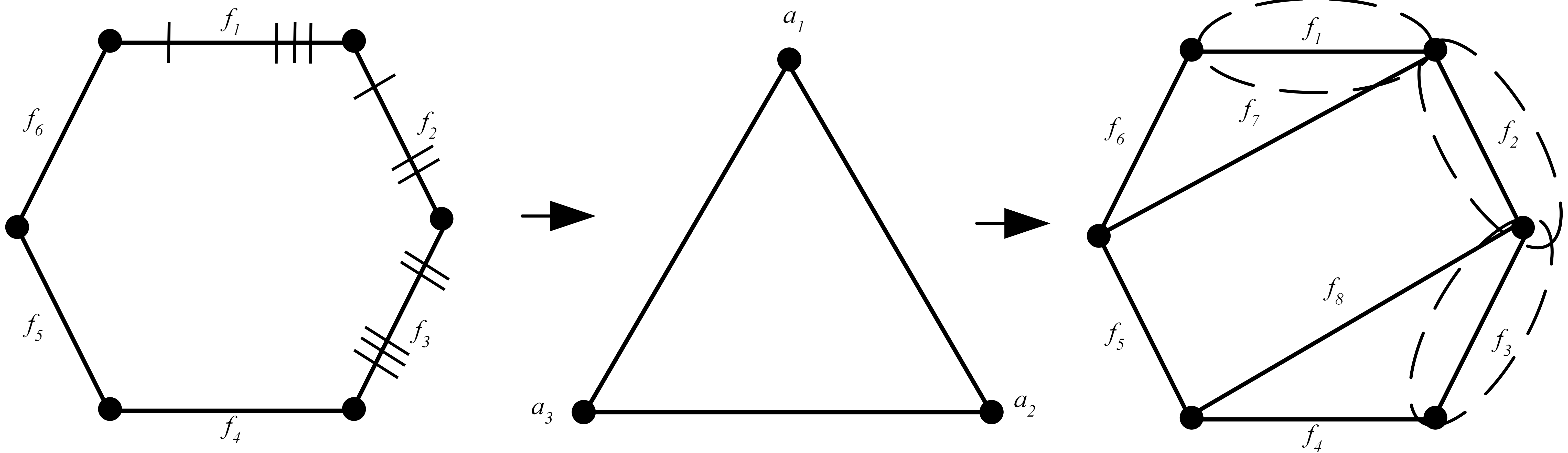
# Конфігурації ПРЦ



Метод перетворення GL-моделей шляхом  
введення внутрішніх ребер з урахуванням  
попарних реберних циклів  
Примак І.К., група КВ-62М



# Блокування ПРЦ в GL-моделі



Метод перетворення GL-моделей шляхом  
введення внутрішніх ребер з урахуванням  
попарних реберних циклів  
Примак І.К., група КВ-62М